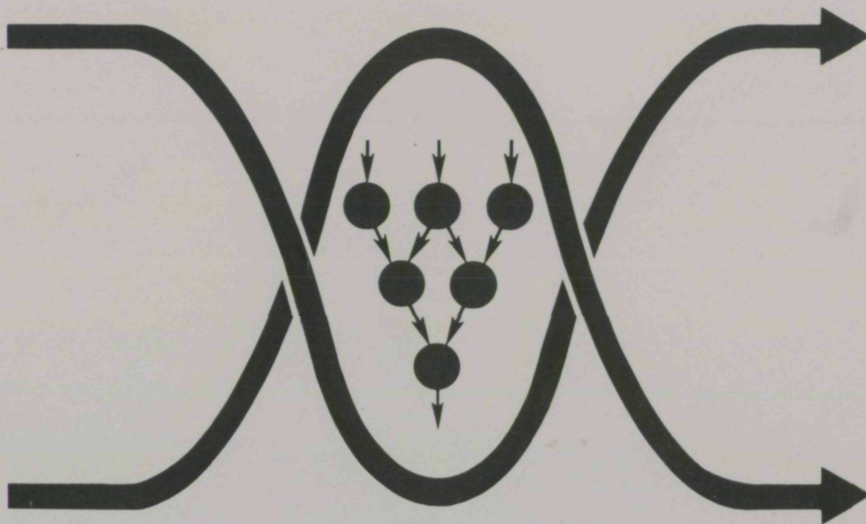
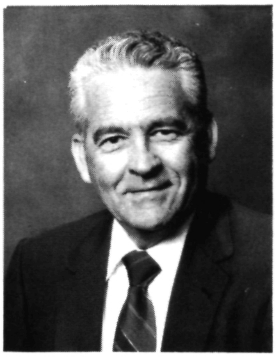


1988 FIRST JOINT TECHNOLOGY WORKSHOP ON NEURAL NETWORKS AND FUZZY LOGIC

Lyndon B. Johnson Space Center
Gilruth Recreation Center
Houston, Texas
May 2-3, 1988

Co-Sponsored by NASA - Johnson Space Center
and the University of Houston - Clear Lake

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH



Message from the General Chair

Within the past few years, we have seen the development and application of expert systems at every NASA center. Today, we have operative expert systems at all the NASA centers and active expert system development at many organizations within each center. The rapid deployment and development of expert systems was the direct result of the availability of hardware and software development support system environments. As dedicated artificial neural systems and fuzzy logic hardware with supporting software become available, application and development systems will proliferate as fast as expert system technology.

The major purpose of this workshop is to provide researchers and practitioners with an opportunity to exchange information in order to determine the requirements and capabilities of technologies necessary to launch these innovative methodologies into mature and productive operative environments.

Robert H. Brown

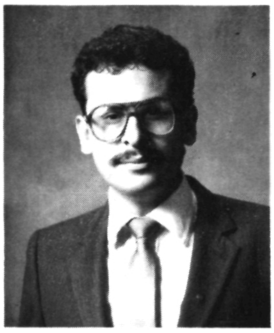
Message from the Technical Chair

As you read the program, you will note that the list of invited speakers is like a "who's who" of the worlds of neural networks and fuzzy logic. We have assembled here an extraordinary group of researchers. I extend to them my deepest appreciation for their willingness to be an integral part of this unique event.

It is a rare opportunity to be at the genesis of an entirely new field of technology. Although the underpinings of this technology have deep roots in the past, we are, at this moment, witnessing the assembling of the critical mass of experimental and theoretical endeavors to bring the fruits of research in neural networks and fuzzy logic into the application domain. I am sure that those of us participating in this workshop will remember it as a significant event in the evolution of our fledgling discipline. I believe that fuzzy logic is vital to our future control systems, and that neural networks are the key to many of the crucial problems that have eluded the present technology. With our combined efforts, we are taking an important step in the quest for the essence of artificial intelligence.

Robert T. Savely

MESSAGE FROM THE TECHNICAL AREA DEVELOPERS



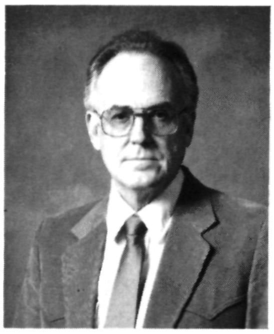
Neural Networks

From a research point of view, artificial neural systems offer many hopes toward synthetically patterning the complex behaviors found in nature. From an applications point of view, artificial neural systems offer hope for a more efficient method of developing control systems, which conventionally are difficult to model. Artificial neural systems are particularly exciting because it is a science which enforces the collaboration of the biological sciences, physical sciences, cognitive sciences, and engineering sciences. This collaboration between sciences is not found elsewhere, and is the basis for a true whole science.

Fuzzy sets, although different from artificial neural systems, also are adapted toward modeling complex systems which were difficult or impossible with conventional methods. The combination of these fields shows great promise toward capturing that elusive problem of modeling nature's complex behaviors.

I would like to extend my sincere appreciation and gratitude to the committee members, the invited speakers, and the participants for making this workshop productive and interesting. I would also like to wish each of you much prosperity and good fortune in the pursuit of these exciting fields.

James Villarreal



Fuzzy Logic

This workshop offers a unique opportunity for scientists and engineers, who have common interests in fuzzy sets and/or neural network technology, to meet and discuss problems of mutual concern. The workshop will feature presentations of current work by noted experts in the above mentioned fields. The areas of discussion will vary widely but will all have commonality in dealing with decisionmaking in environments where knowledge is imprecise, vague, incomplete and exact models are impossible or at least impractical to build.

The fields of fuzzy sets and neural networks, although considerably different in approach to problem solutions, seem to have some common applicability areas, or, for example, in visual and voice pattern recognition. It is hoped that at this workshop much valuable information will be exchanged and some problems of mutual interest in both fuzzy sets and neural networks will surface.

We are indeed grateful to the prominent scientists who have agreed to provide their time and share their ideas with us. Without them, this workshop could not be successful. However, we also wish to acknowledge the fact that a successful workshop atmosphere depends on the communication and interaction of all attendees. We sincerely appreciate everyone's participation and support.

Robert N. Lea

Neural Network/Fuzzy Logic Program Overview

Monday, May 2

7:30	Registration
8:00-8:30	Welcome and Introduction

OPENING ADDRESS SPEAKERS:

8:30-9:30	Stephen Grossberg - Boston University *Emergent Invariants of Self-organizing Neural Networks for Pattern Recognition and Robotics*
9:30-10:30	Bart Kosko - University of Southern California *Fuzzy Theory and Neural Networks*
10:30-11:30	Lotfi Zadeh - University of California *The Role of Fuzzy Sets in the Treatment of Uncertainty in Control Processes and Knowledge Representation*

SESSION 1

1:00-1:30	Takeshi Yamakawa - Kumamato University *A Fuzzy Microprocessor: A Novel Device for High-Speed Approximate Reasoning*
1:30-2:00	Masaki Togai - Togai Infralogic, Inc. *Fuzzy and Neural Net Processor and its Programming Environment*
2:00-2:30	Hiroyuki Watanabe - University of North Carolina *Fuzzy Logic Inference Processor: Custom VLSI Design for System Integration*
2:30-3:00	Kaoru Hirota - Hosei University *An Application of Fuzzy Logic to Robotic Vision and Control*

SESSION 2

3:15-3:45	Demetri Psaltis *Optical Neural Computers*
3:45-4:15	Harold Szu - Naval Research Laboratory *What is the Significance of Neural Networks for AI?*
4:15-4:45	Daniel Levine - University of Texas at Arlington *Neural Modeling of Selective Attention*

KEYNOTE DINNER

7:00-8:00	**COCKTAIL RECEPTION**
8:00-9:30	**DINNER - DR. LEON COOPER - KEYNOTE SPEAKER**

Tuesday, May 3

SESSION 3

8:30-9:00 James Bower - California Institute of Technology
Applied and Real Neural Networks: A Coordinated and Interdependent Investigation of Both

9:00-9:30 Walter Freeman - University of California, Berkeley
Implementation of Pattern Recognition Algorithms Derived from Olfactory Information Processing

9:30-10:00 Guenter Gross - North Texas State University
Multi-electrode Burst Pattern Feature Extraction from Mammalian Networks in Culture

10:00-10:30 Mike Myers - TRW ANS Center
A Hybrid Connectionist - AI Architecture for Reflective and Exploratory Systems

SESSION 4

10:45-11:15 William Siler - Mote Marine Laboratory
Applications of a Fuzzy Expert System

11:15-11:45 Maria Zemankova - University of Tennessee
Intelligent Information Systems with Learning Capabilities

SESSION 5

1:15-1:45 Pentti Kanerva - RIACS, NASA Ames Research Center
Understanding Information - processing in Animals as a Way to Building Intelligent Robots

1:45-2:15 Claude Cruz - Plexus Systems
Knowledge Processing Using Neural Networks

2:15-2:45 Rod Taber - General Dynamics
Fuzzy Logic Operators and Neuron Activation Fields

2:45-3:15	Douglas Reilly - Nestor, Inc. *Adaptive Pattern Recognition Using a Multi-Neural Network Learning System*
3:30-4:00	James Bezdek - Boeing *Knowledge Representation by Linguistic Transitive Closures of Trapezoidal Fuzzy Members*
4:00-4:30	Bill Buckles - Tulane University *Relationship Between Uncertainty and Databases and Expert Systems*
4:30-5:00	James Buckley - University of Birmingham *Linear Fuzzy Controller*

* Busing will be provided for lunch hours only and will depart from the Gilruth Recreation Center, arriving at Building 11 cafeteria for lunch. Buses will pick those eating at cafeteria up and return them to the Gilruth at specified times.

Monday -	11:45 am departure 12:45 pm arrival
Tuesday -	12:00 noon departure 1:00 pm arrival

Neural Networks/Fuzzy Logic Committee Members

General Chair
Robert H. Brown, NASA/JSC

Technical Chair
Robert T. Savely, NASA/JSC

Technical Area Developers

Artificial Neural Systems
James A. Villarreal, NASA/JSC

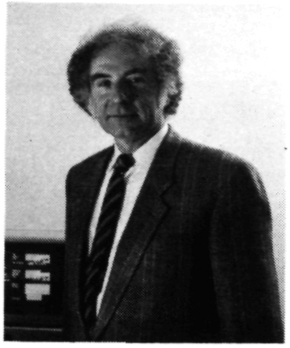
Fuzzy Logic
Robert N. Lea, NASA/JSC

Executive Chair
Sandy Griffin, NASA/JSC

Assistant Executive Chair
Carla Armstrong, NASA/Barrios

Administrative Chair
Carol Kasworm, U of H/Clear Lake

Local Publicity Chair
Daniel C. Bochsler, NASA/Lincom



Keynote Speaker

Leon Cooper, Ph.D.

Dr. Cooper is cofounder and chairman of the board of Nestor, Inc. He received his A.B. in 1951, his A.M. in 1953, and a Ph.D. in 1954, all from Columbia University. He has been a professor, an associate professor, or a visiting professor at various universities and summer schools. He has also served as a consultant for various governmental agencies, and industrial and educational organizations. Dr. Cooper has given various public lectures and presented papers at international conferences and symposia. Throughout his career, he has been the recipient of the following fellowships and awards:

Nobel Prize (with J. Bardeen and J. R. Schrieffer), 1972

Award of Excellence, Graduate Faculties Alumni of Columbia University, 1974

Descartes Medal, Academie de Paris, Universite Rene Descartes, 1977

John Jay Award, Columbia College, 1985

Who's Who, Who's Who in America, Who's Who in the World, various other listings

Comstock Prize (with J. R. Schrieffer), National Academy of Sciences, 1968

NSF Postdoctoral Fellow, 1954-55

Alfred P. Sloan Foundation Research Fellow, 1959-66

John Simon Guggenheim Memorial Foundation Fellow, 1965-66

Fellow, American Physical Society, American Academy of Arts and Sciences

Member, American Philosophical Society; National Academy of Sciences; Sponsor Federation of American Scientists; Society for Neuroscience; American Association for Advancement of Science; Institute for Advanced Study, 1954-55; Counseil Superieur de la Recherche de l'Universite Rene Descartes (Academie de Paris, Paris V); Phi Beta Kappa; Sigma Xi

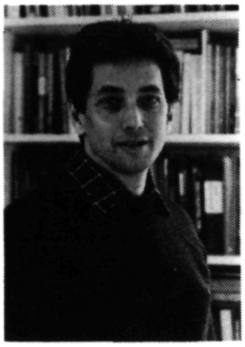
Doctor of Sciences (honoris causa), Columbia University, 1973; University of Sussex, 1973; University of Illinois, 1974; Brown University, 1974; Gustavus Adolphus College, 1975; Ohio State University, 1976; Universite Pierre et Marie Curie, Paris, 1977

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

51

S1-63

N91-71351 15302 17



Stephen Grossberg, Ph.D.

Boston University
Boston, Massachusetts

Dr. Grossberg received his graduate training at Stanford University and Rockefeller University, and was a professor at the Massachusetts Institute of Technology before assuming his present position at Boston University. He is professor of mathematics, psychology, and biomedical engineering at Boston University, where he founded and is the director of the Center for Adaptive Systems. He is also the director of the university's new graduate program in cognitive and neural systems. In addition, Dr. Grossberg is president of the International Neural Network Society and coeditor-in-chief of the society's journal, Neural Networks. During the past few decades, Dr. Grossberg and his colleagues at the Center for Adaptive Systems have pioneered and developed a number of the fundamental principles, mechanisms, and architectures that form the foundation for contemporary neural network research.

EMERGENT INVARIANTS OF SELF-ORGANIZING NEURAL NETWORKS FOR PATTERN RECOGNITION AND ROBOTICS

Abstract

Described are several real-time neural network architectures that are capable of self-organizing invariant behavioral properties in applications to sensory pattern recognition, cognitive information processing, and adaptive sensory-motor control. These invariants include a similarity invariant that arises in adaptive pattern recognition and cognitive information processing; a position invariant that arises in determining the location of a target with respect to the head; and a synchrony invariant that enables motor systems with multiple degrees of freedom, such as arms and speech articulators, to generate flexible and synergetic planned movements.

EMERGENT INVARIANTS OF SELF-ORGANIZING NEURAL NETWORKS FOR PATTERN RECOGNITION AND ROBOTICS

Stephen Grossberg

A lecture delivered at
The 1988 First Joint Technology Workshop
on Neural Networks and Fuzzy Logic

May 2, 1988
Lyndon B. Johnson Space Center
Houston, Texas

EMERGENT INVARIANTS

A KEY PROBLEM OF INTELLIGENT BEHAVIOR

SELF-ORGANIZED

DISCOVERY

LEARNING

PRODUCTION

OF INVARIANT BEHAVIORAL PROPERTIES BY A REAL-TIME NEURAL NETWORK

MANY DIFFERENT TYPES

INVARIANT	BEHAVIOR
SIMILARITY	RECOGNITION
POSITION	TARGETING
SYNCHRONY	MULTI-JOINT MOVEMENT

SIMILARITY INVARIANT RECOGNITION

HOW DOES A NEURAL NETWORK LEARN TO RECOGNIZE SIMILAR PATTERNS AS EXEMPLARS OF A SINGLE CATEGORY?

ADAPTIVE RESONANCE THEORY (1976)
A R T

1. EXPLAIN AND PREDICT
BEHAVIORAL AND NEURAL DATA

PSYCHOLOGICAL REVIEW, 1980
1982
1986

STUDIES OF MIND AND BRAIN,
REIDEL, 1982

THE ADAPTIVE BRAIN, VOLS. I + I
ELSEVIER/NORTH-HOLLAND, 198

2. MATHEMATICAL + COMPUTATIONAL
ANALYSES
ARCHITECTURE DEVELOPMENT

GAIL CARPENTER, ART 1 + 2

ADAPTIVE RESONANCE THEORY

STABLE SELF-ORGANIZATION OF RECOGNITION CODES FOR ARBITRARY SEQUENCES OF ANALOG OR DIGITAL INPUT PATTERNS.

(GAIL CARPENTER)

ART 1 - DIGITAL WORLD

(CVGIP, JAN., 1987)

ART 2 - ANALOG OR DIGITAL WORLD
(APPLIED OPTICS, NOV., 1987)

WHY DO WE PAY ATTENTION?

WHY DO WE LEARN EXPECTATIONS ABOUT THE WORLD?

HOW DO WE COPE SO WELL WITH UNEXPECTED EVENTS?

— WHEN WE ARE ON OUR OWN, WITHOUT A TEACHER?

HOW DO WE KNOW WHAT COMBINATIONS OF FACTS

ARE PREDICTIVE?
IRRELEVANT?

HOW DO WE QUICKLY RECOGNIZE FAMILIAR FACTS

WITHOUT HAVING TO SEARCH EVERYTHING ELSE THAT WE KNOW

MAIN IDEA

TOP-DOWN ATTENTIVE FEEDBACK
ENCODES

LEARNED EXPECTATIONS
THAT

SELF-STABILIZE LEARNING
IN RESPONSE TO

ARBITRARY TEMPORAL SEQUENCE
OF INPUT SPATIAL PATTERNS

IN

REAL-TIME

ATTENTIVE INFORMATION PROCESSING $\rightleftarrows$ LEARNING AND MEMORY

4 TYPES OF ATTENTIONAL MECHANISM

ATTENTIONAL PRIMING

ATTENTIONAL GAIN CONTROL

ATTENTIONAL VIGILANCE

INTERMODAL COMPETITION

ART

ARCHITECTURE RESOLVES KEY DESIGN TRADE-OFFS

STABILITY-PLASTICITY DILEMMA

HOW DOES A REAL-TIME SYSTEM SWITCH BETWEEN ITS STABLE AND PLASTIC MODES WITHOUT AN EXTERNAL TEACHER?

HOW CAN IT BE PLASTIC TO IMPORTANT EVENTS

AND STABLE TO IRRELEVANT EVENTS?

TOO STABLE	TOO PLASTIC
↓	↓
RIGID	CHAOTIC

SYSTEMS THAT ARE SENSITIVE TO

NOVELTY

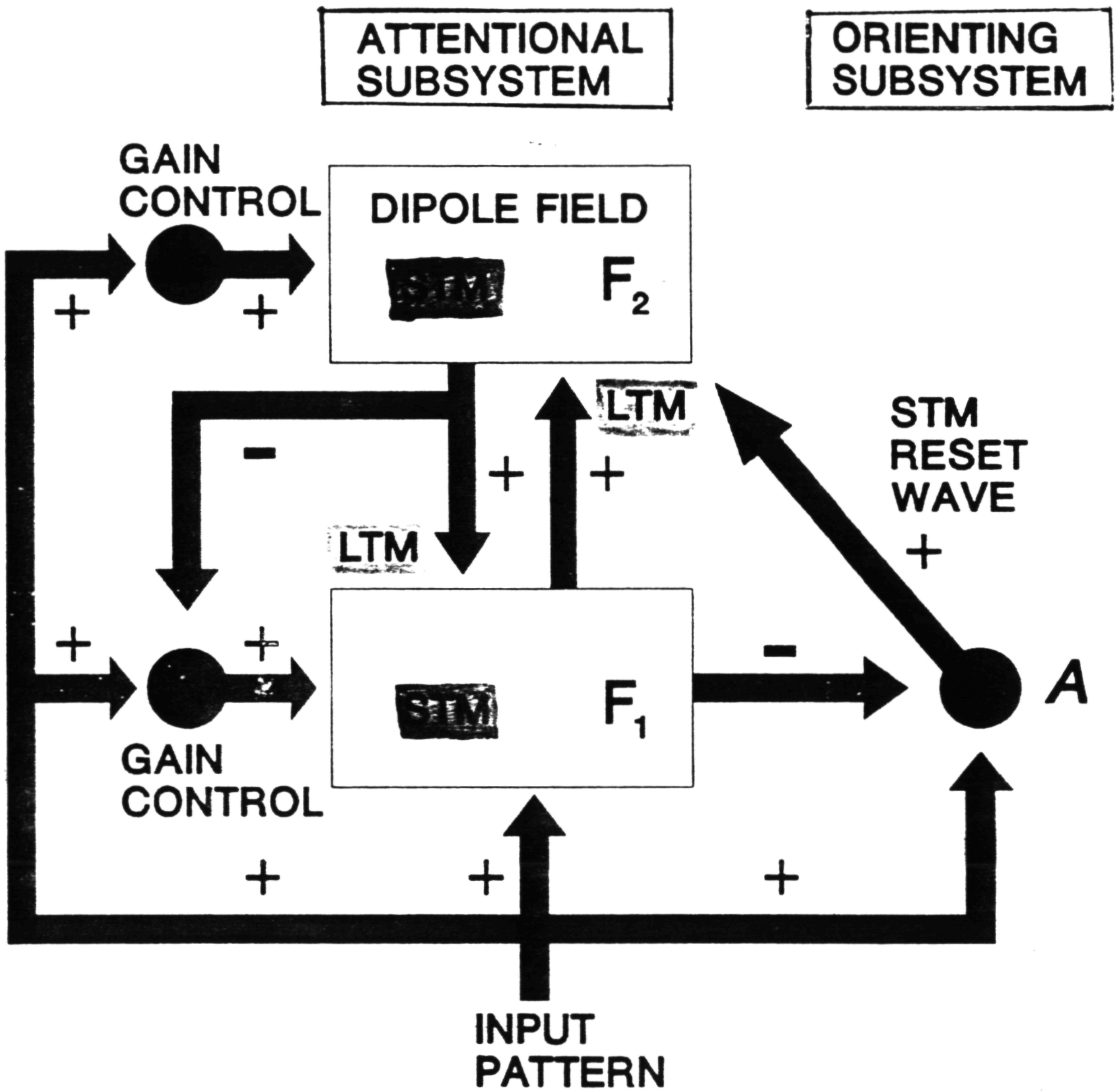
UNEXPECTED EVENTS RAPIDLY REORGANIZE INFORMATION PROCESSING NOT "JUST" MATCHING!

ART

MULTIPLE INTERACTING MEMORY SYSTEMS

ATTENTIONAL SUBSYSTEM	⟷	ORIENTING SUBSYSTEM
EXPECTED EVENTS		UNEXPECTED EVENTS
FAMILIAR EVENTS		UNFAMILIAR EVENTS

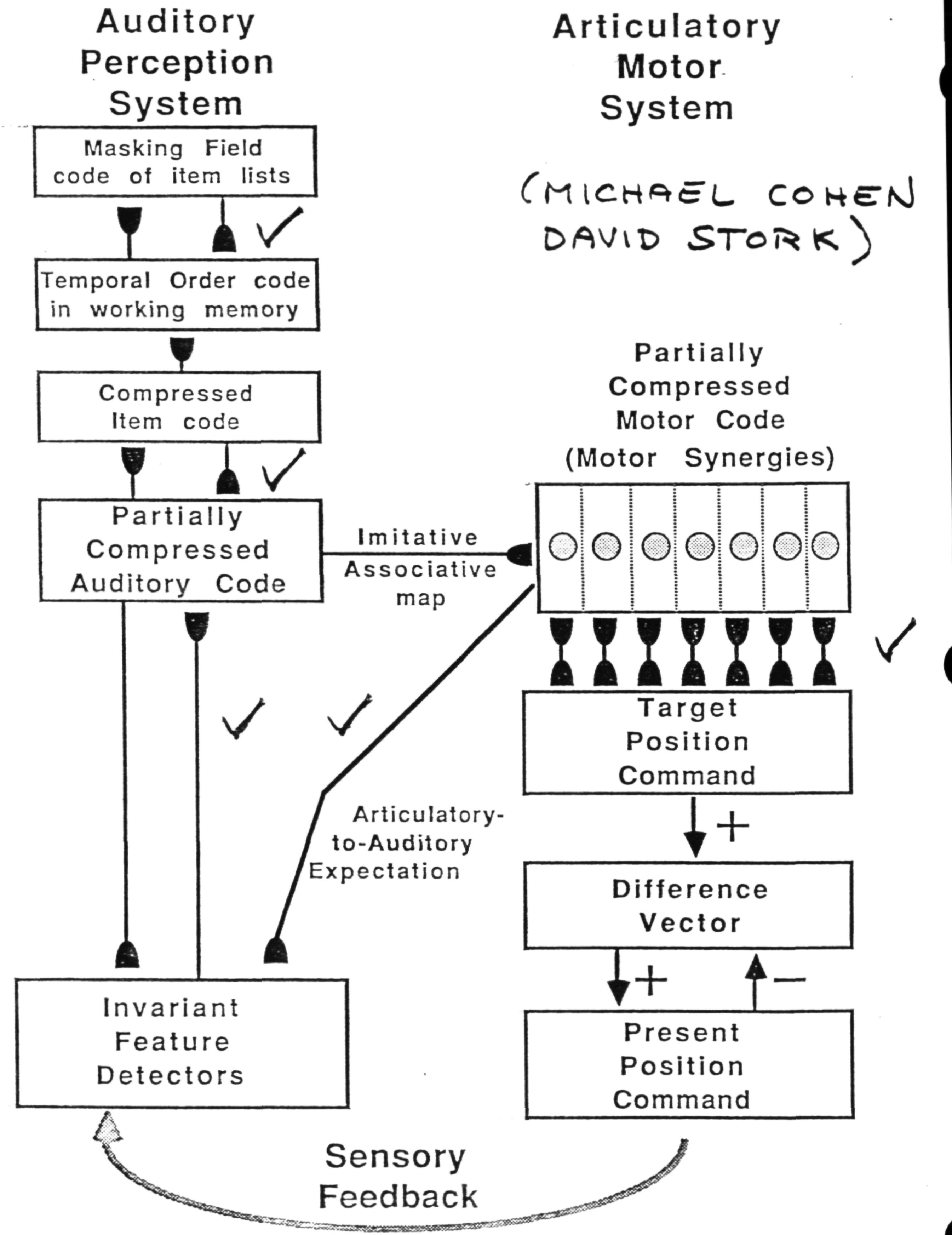
ART 1 ARCHITECTURE



NONSPECIFIC CONTROL STRUCTURES

NEURAL MODULATORS
AROUSAL

SPEECH PERCEPTION + PRODUCTION



61

ADAPTIVE RESONANCE SYSTEM	AUTOASSOCIATORS BOLTZMANN MACHINE BACK-PROPAGATION
SELF-ORGANIZE BUILDS INTERNAL CODES AND EXPECTATIONS	EXTERNAL TEACHER SOURCE OF PRE-CODED EXPECTED OUTPUT
SELF-STABILIZE WORLD KEEPS GOING INTERNAL MEMORY BUFFER	CAPACITY CATASTROPHE CODE WASHES AWAY WITH TOO MANY INPUTS SHUT OFF WORLD!
USE FULL CAPACITY	CAN'T USE FULL CAPACITY $\sim .15n$
MAINTAIN PLASTICITY RETAIN POTENTIAL FOR NEW LEARNING INDEFINITELY UNEXPECTED INPUTS	TERMINATE LEARNING TOO SOON? TOO LATE? JUST RIGHT! OMNISCIENT TEACHER
LEARN IN APPROXIMATE-MATCH PHASE BUFFERED AGAINST FREQUENT NOISE LOCAL MINIMA? NO!	LEARN IN MISMATCH PHASE WASHED AWAY BY FREQUENT NOISE STEEPEST DESCENT

HOW TO AVOID
SPURIOUS MEMORY STATES
LOCAL MINIMA?

USE NOISE 
EXTERNAL PARAMETERS $(T \to T_c$

CRITICAL SLOWING DOWN

NOT REAL-TIME!

TOO DIFFUSE!

ART SOLVES ANOTHER
TRADE-OFF

$$\boxed{\left\{\begin{array}{c}\text{ADAPTIVE}\\ \text{SEARCH}\end{array}\right\} - \left\{\begin{array}{c}\text{RECOGNITION}\\ \text{SPEED}\end{array}\right\}}$$

<u>SELF-ADJUSTING PARALLEL</u>
<u>MEMORY SEARCH</u>
ACTIVELY REORGANIZES
ENERGY LANDSCAPE
AS IT QUICKLY
<u>TESTS HYPOTHESES</u>
ABOUT THE WORLD

ART	
SELF-ADJUSTING MEMORY SEARCH REMAINS EFFICIENT IN ARBITRARY ENVIRONMENT AT ARBITRARY STAGE OF LEARNING	SEARCH TREE FIXED
DIRECT ACCESS NO SEARCH AS RECOGNITION INVARIANTS BECOME FAMILIAR SEARCH MECHANISM AUTOMATICALLY DISENGAGES	SEARCH TIME INCREASES WITH CODE COMPLEXITY LOG (n)

TIME TAKEN TO RECOGNIZE YOUR PARENTS AT

AGE 5 ?
20 ?
35 ?

LEARN IN APPROXIMATE-MATCH STAGE AT END OF SEARCH ON EACH TRIAL

ACCESS FAMILIAR RECOGNITION CODE	REFINE IT BASE ON NEW INFORMAT
ACCESS UNCOMMITTED RECOGNITION CODE	START A NEW CODE

HOW YOU MATCH

DETERMINES WHAT YOU CAN

STABLY LEARN

FAST INFORMATION PROCESSING (STM)

REAL-TIME!

LEARNING (LTM)

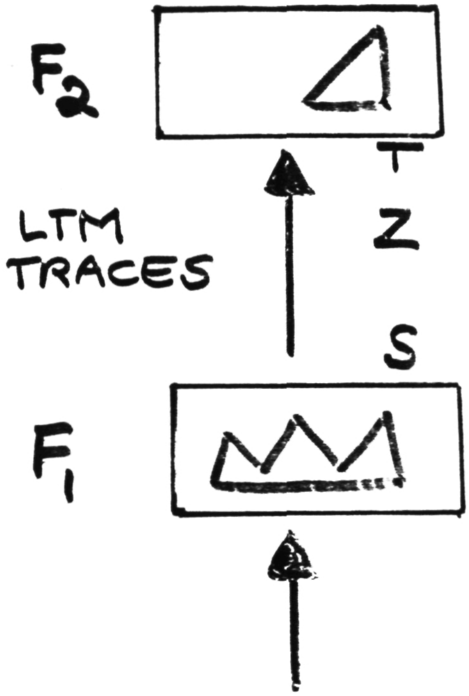
INTENTIONALITY!

(AI VS. SEARLE)

SPATIAL LOGIC

BASIC CODING STRUCTURE

COMPETITIVE LEARNING



COMPRESSED STM CODE, OR REPRESENTATION

$T = ZS$ ADAPTIVE FILTER

STM ACTIVITIES ACROSS A FIELD OF FEATURE DETECTORS

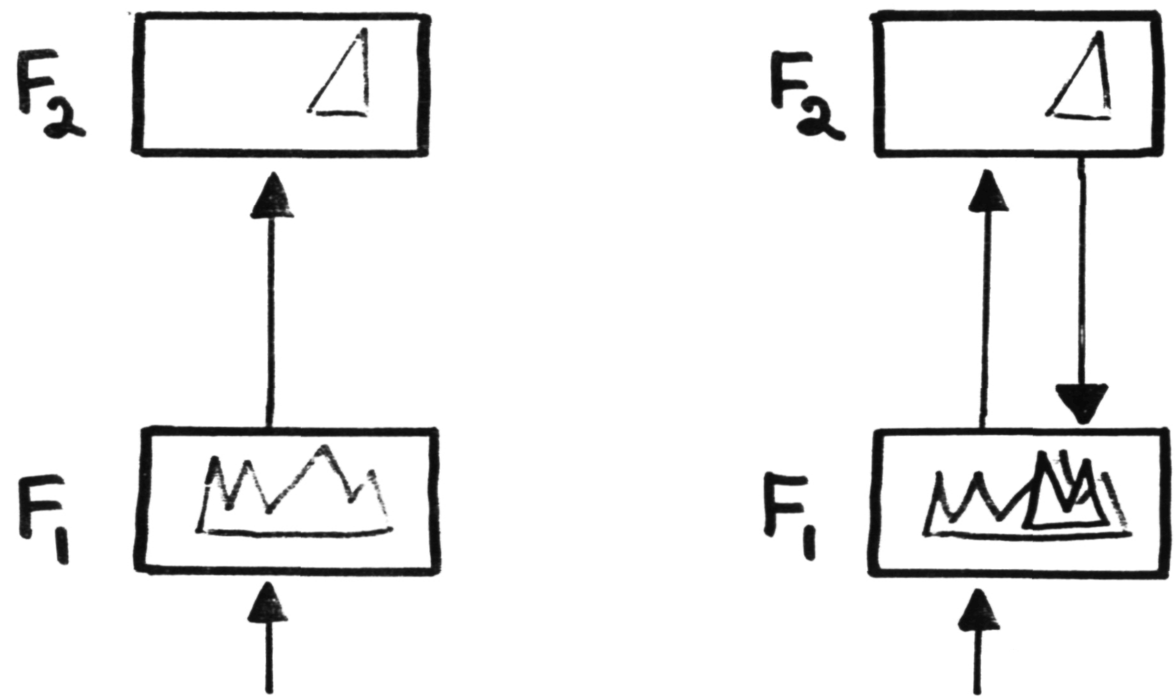
PREPROCESSED INPUTS

THEOREM (1976)
SUCH A SYSTEM CAN STABLY LEARN A RECOGNITION CODE IF THE INPUT PATTERNS ARE NOT TOO NUMEROUS OR DENSE.

IN GENERAL, **UNSTABLE!** ⇒ **ART.**

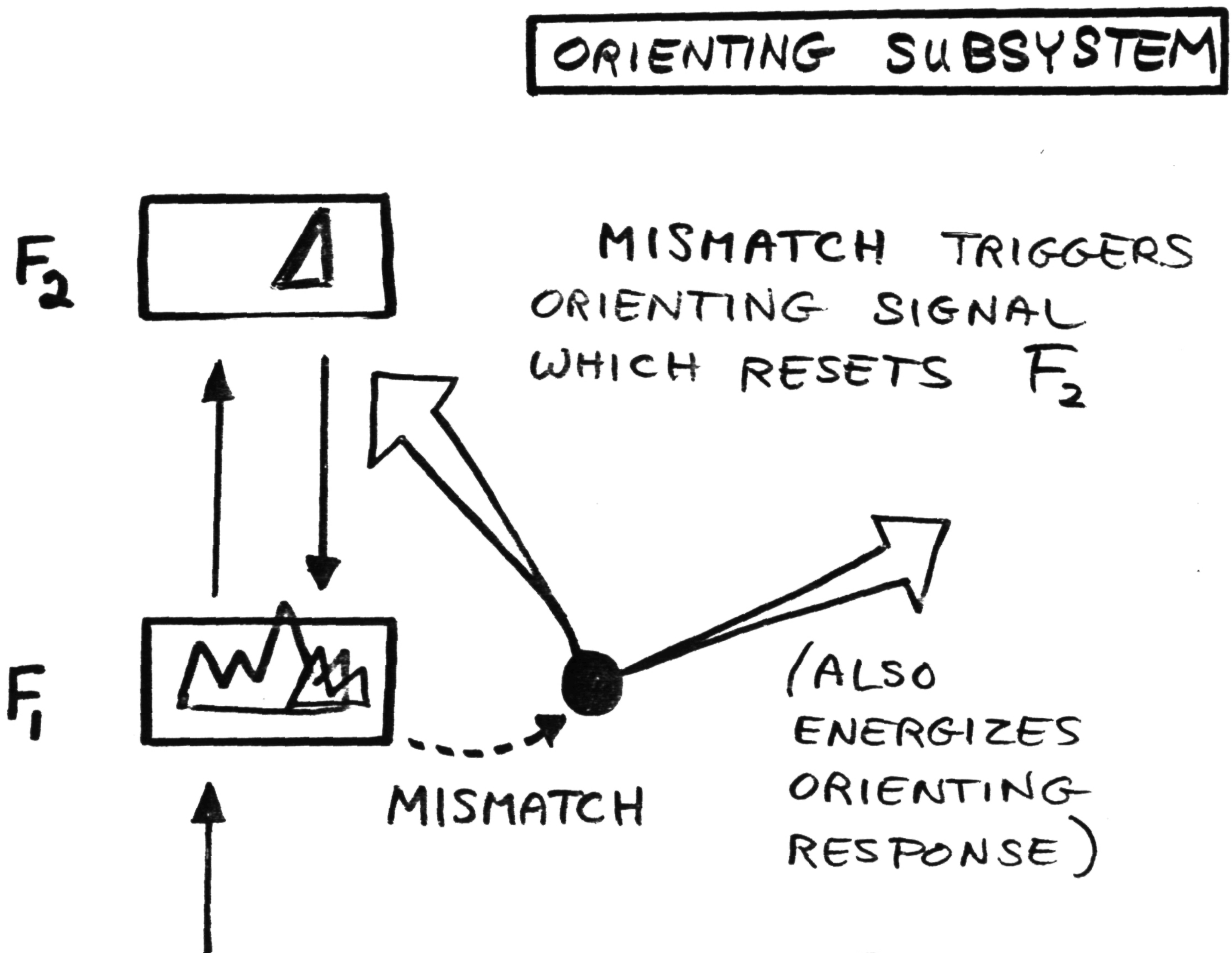
RECENT LEARNING CAN WASH AWAY OLDER LEARNING

LEARNED TOP-DOWN EXPECTANCIES, OR {PROTOTYPES / TEMPLATES}, CAN STABILIZE LEARNING IN RESPONSE TO AN ARBITRARY SEQUENCE OF PATTERNS



HOW? { MATCHING AT F_1 OF BU AND TD PATTERNS STABILIZES LEARNING

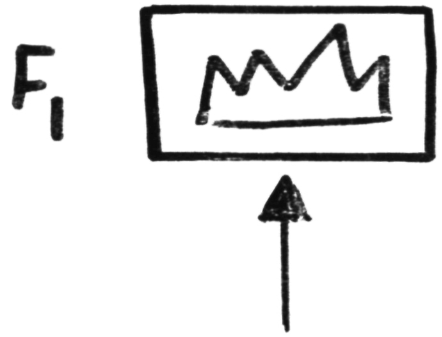
A "BIG ENOUGH" MISMATCH AT F_1 QUICKLY RESETS THE F_2 CODE BEFORE NEW LEARNING CAN OCCUR



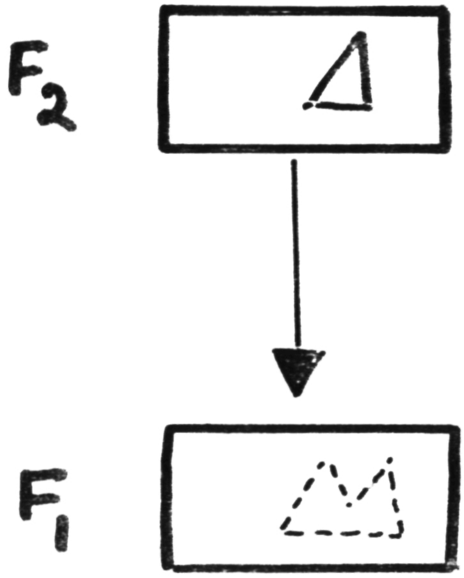
WHEN?

HOW TO MATCH BU AND TD AT F_1?

RECONCILE 2 REQUIREMENTS



1. AUTOMATIC REGISTRATION OF BU INPUT PATTERNS



2. TD PRIME CAN BE SUBLIMINAL

EXPECTANCY

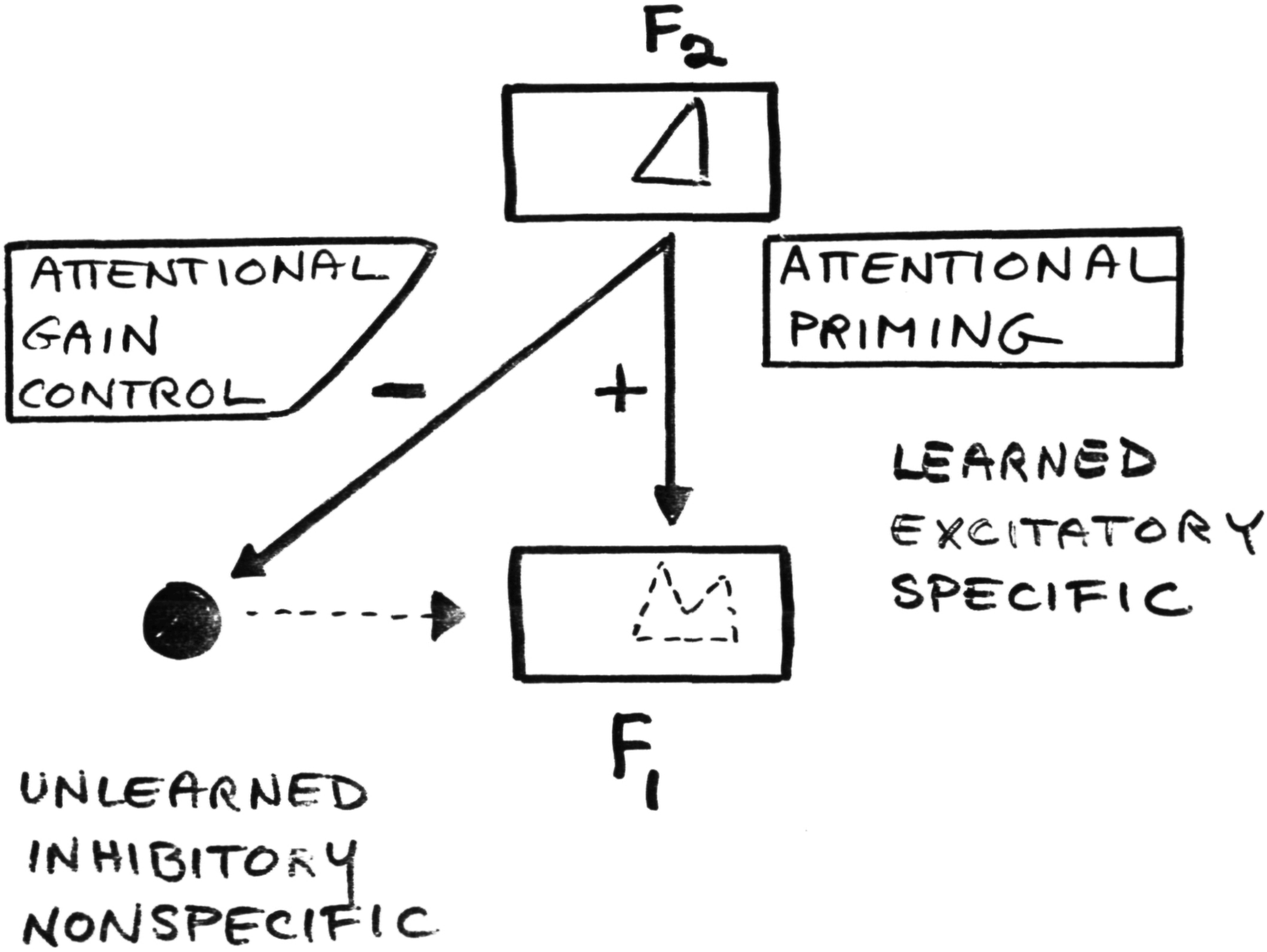
INTENTIONALITY

SPATIAL LOGIC

INTENTIONALITY → LOGIC

HOW CAN F_1 KNOW THE DIFFERENCE BETWEEN BU AND TD SIGNALS?

FUNDAMENTAL DISTINCTION



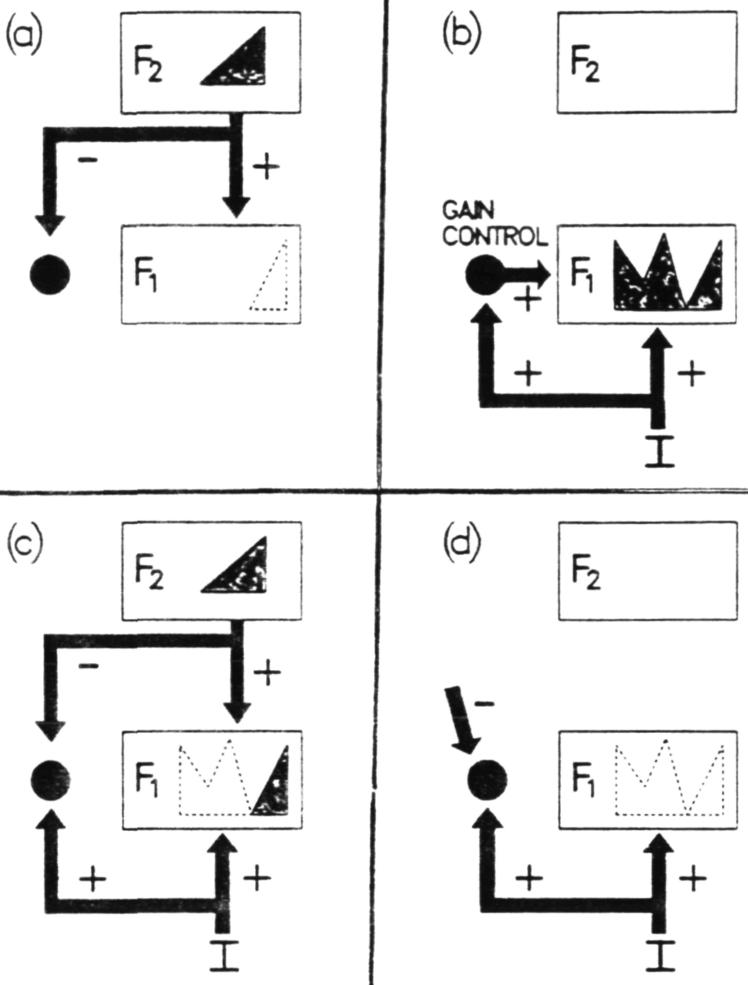
ATTENTIONAL GAIN CONTROL ⇒ MATCHING RUL

$\frac{2}{3}$ RULE

TD PRIME

BU INPUT

TD MATCH BU



"SPATIAL LOGIC"

INTERMODAL COMPETITION

INTENTIONALITY ~ LOGIC

$\frac{2}{3}$ RULE MATCHING

IS NECESSARY FOR

STABLE LEARNING

GIVEN ARBITRARY INPUTS.

HOW YOU <u>MATCH</u>
DETERMINES
WHAT YOU CAN STABLY <u>LEARN</u>

$\frac{2}{3}$ RULE +

WEBER LAW RULE
ASSOCIATIVE DECAY RULE } NON-HEBBIAN ASSOCIATIVE LEARNING RULE (LTM)

VIGILANCE RULE } RESET (STM)

ALL RULES EMERGE AS PROPERTIES OF NETWORK INTERACTIONS

NO EXTERNAL PROGRAM

NO PREWIRED SEARCH STRATEGY

SHOW HOW "INDEX" OF CATEGORY CAN REMAIN INVARIANT AS THE CATEGORICAL CRITERIA ARE REFINED.

NON-HEBBIAN ASSOCIATIVE LAW

STM TRACE

SIGNAL

LTM TRACE

GATED LEARNING AND MEMORY DECAY

$$\frac{d}{dt} z_{ij} = \epsilon f(x_j)\left[-z_{ij} + S_i\right]$$

STEEPEST DESCENT

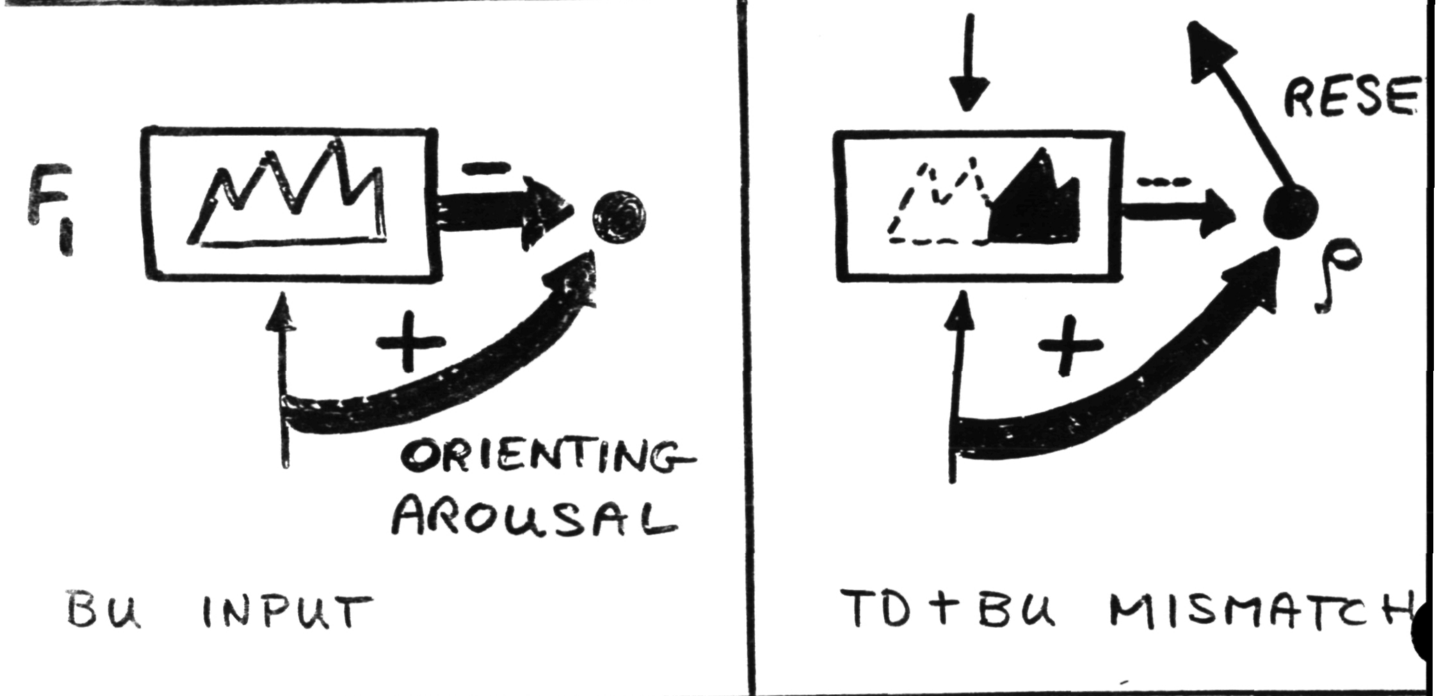
THEORY: GROSSBERG, 1969

EXPERIMENTS: RAUSCHECKER + SINGER, 1979

LEVY et al, 1983.

VIGILANCE RULE:

F_1 MISMATCH $\Rightarrow$ F_2 RESET



ORIENTING SUBSYSTEM

1. $\frac{2}{3}$ RULE $\Rightarrow$ TD CAN SUPPRESS A PORTION OF F_1 STM PATTERN

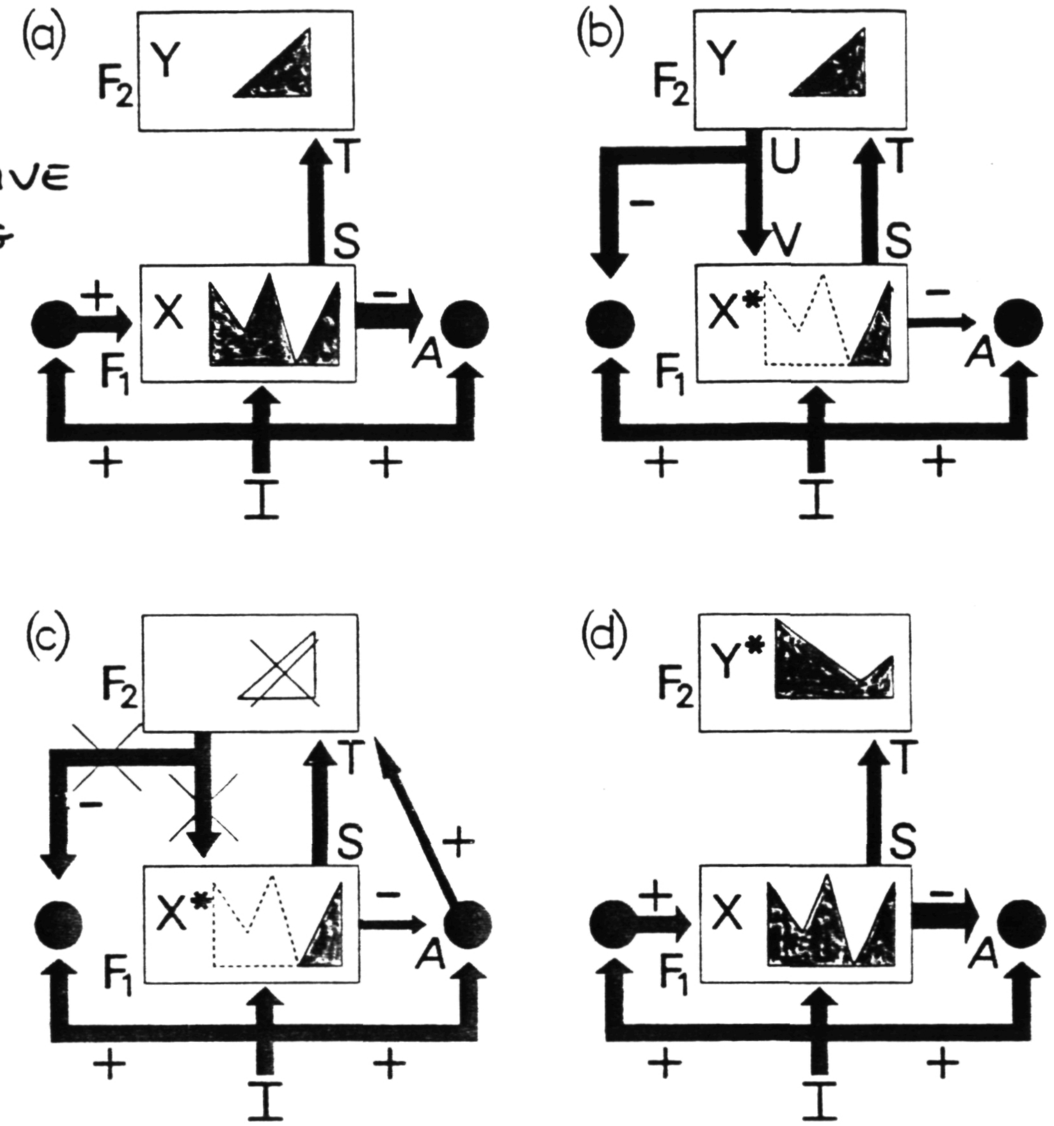
2. F_2 RESET IF

DEGREE OF MATCH $< \rho =$

VIGILANCE PARAMETER

ART 1 SEARCH CYCLE

COMPETITIVE LEARNING (1969-76)

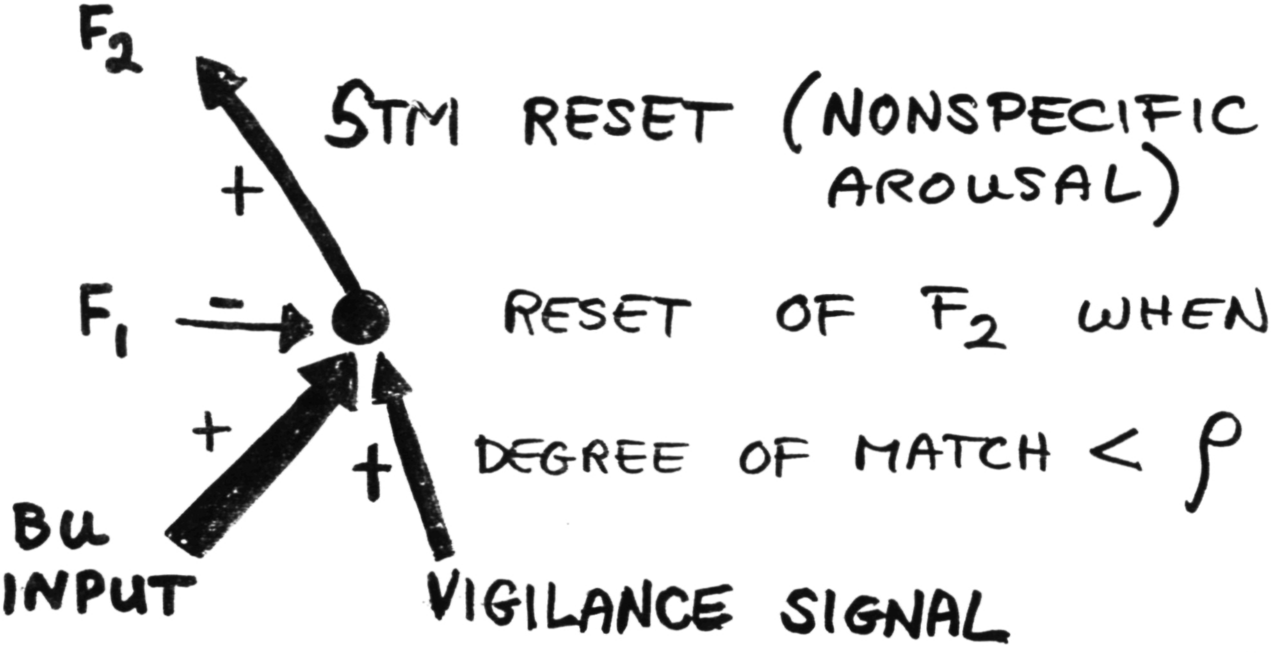


DIRECT ACCESS = NO RESET THEOREM:

DEEP FACT: SYSTEM CONVERGES TO THE NO RESET CASE IN RESPONSE TO AN ARBITRARY LIST OF INPUTS.

(CARPENTER + GROSSBERG, CUGIP, JAN. 1987)

VIGILANCE PARAMETER ρ



LOW VIGILANCE $\Rightarrow$ COARSE CATEGORIES

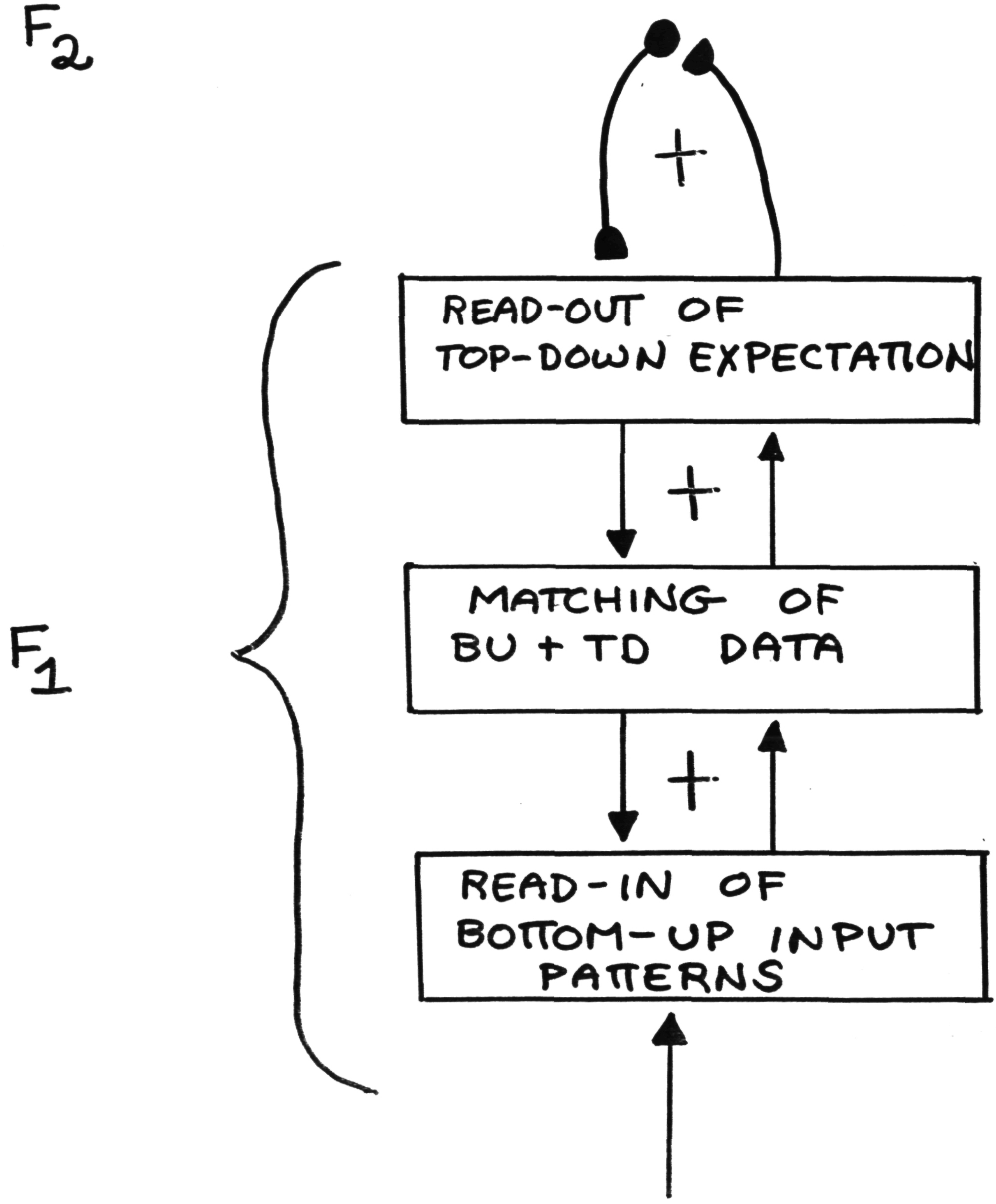
HIGH VIGILANCE $\Rightarrow$ FINE CATEGORIES

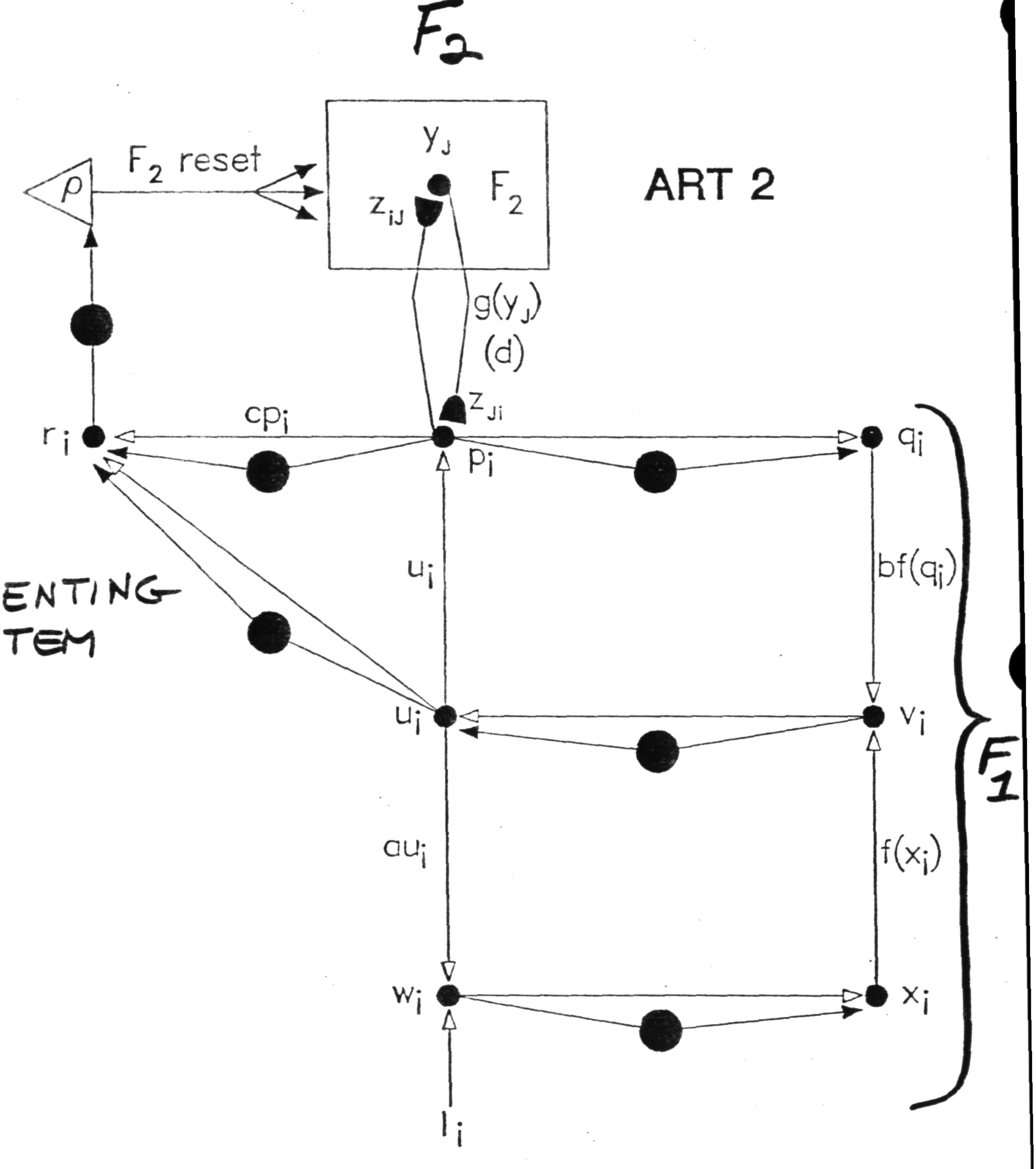
IF BEHAVIORAL FAILURE (e.g., PUNISHMENT) $\Rightarrow$ INCREASED VIGILANCE ($\rho\uparrow$),

SYSTEM AUTOMATICALLY LEARNS FINER CATEGORIES

eg ESKIMO NAMES FOR BLUE SKY.

STM INVARIANCE UNDER READ-OUT OF MATCHED TOP-DOWN LTM

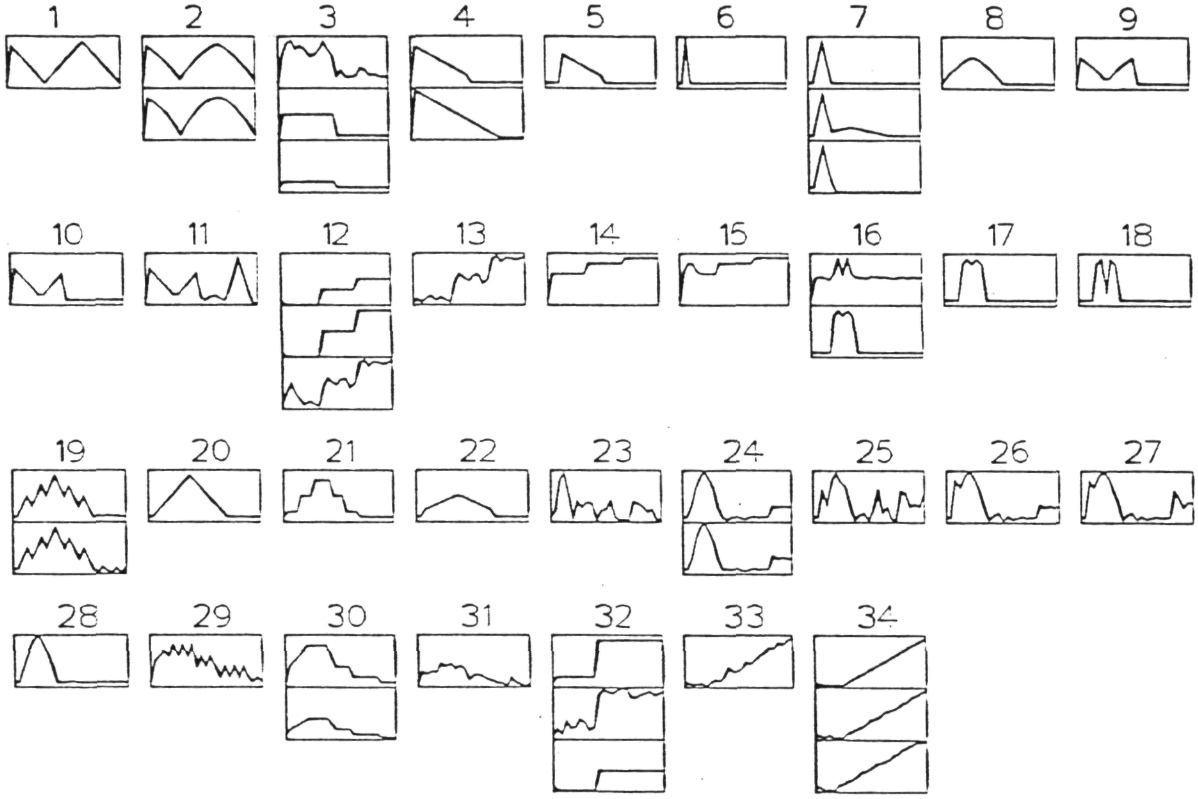




F2
ART 2
F2 reset
ρ
yJ
F2
ziJ
g(yJ)
(d)
zJi
cpi
ri
pi
qi
ui
bf(qi)
ORIENTING
SYSTEM
ui
vi
F1
aui
f(xi)
wi
xi
Ii

ART 2 GROUPING OF 50 INPUT PATTERNS INTO 34 RECOGNITION CATEGORIES ON 1 LEARNING TRIAL

FAST!



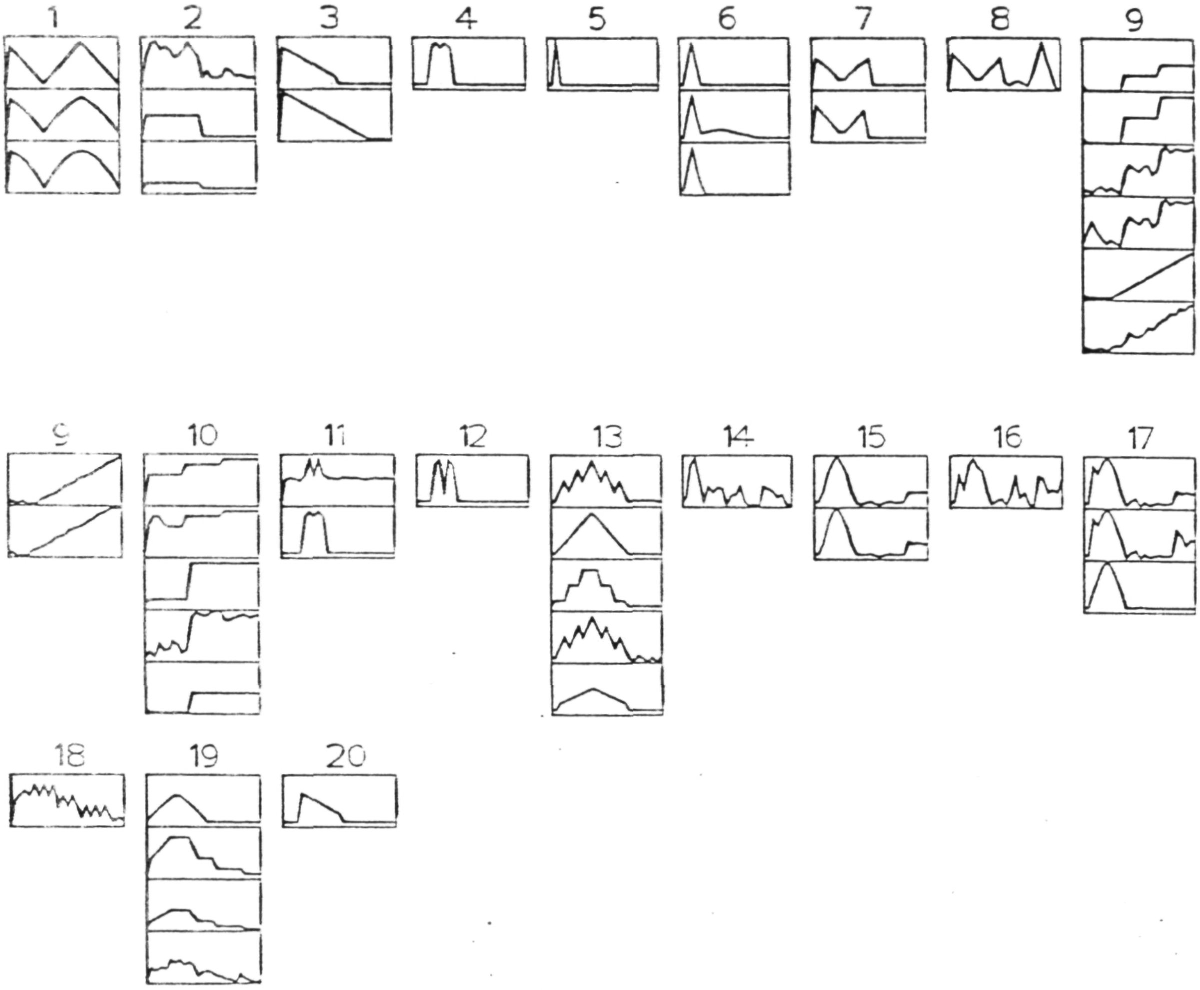
THEREAFTER

STABLE CATEGORIES

DIRECT ACCESS

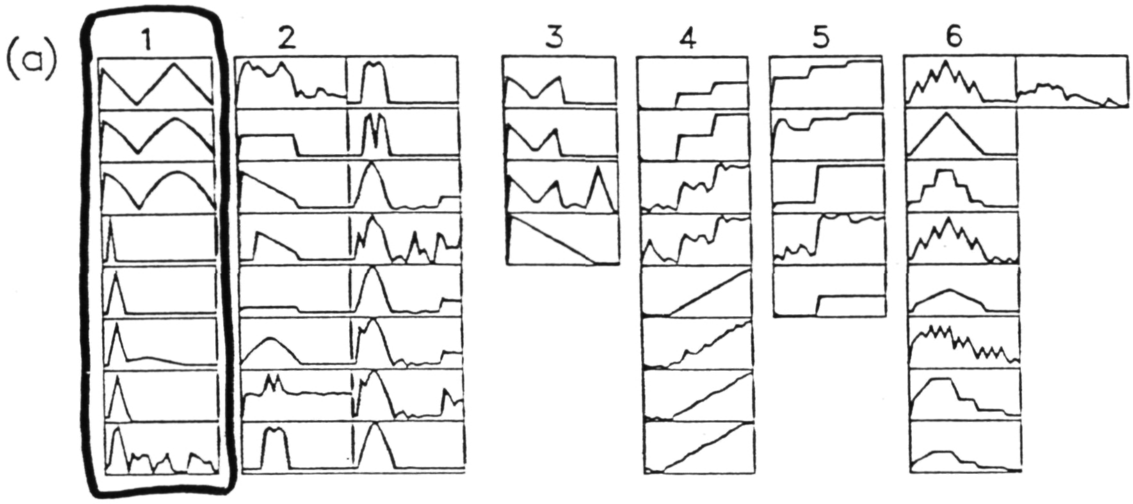
NO SEARCH

LOWER VIGILANCE - 20 INSTEAD OF 34 CATEGORIES

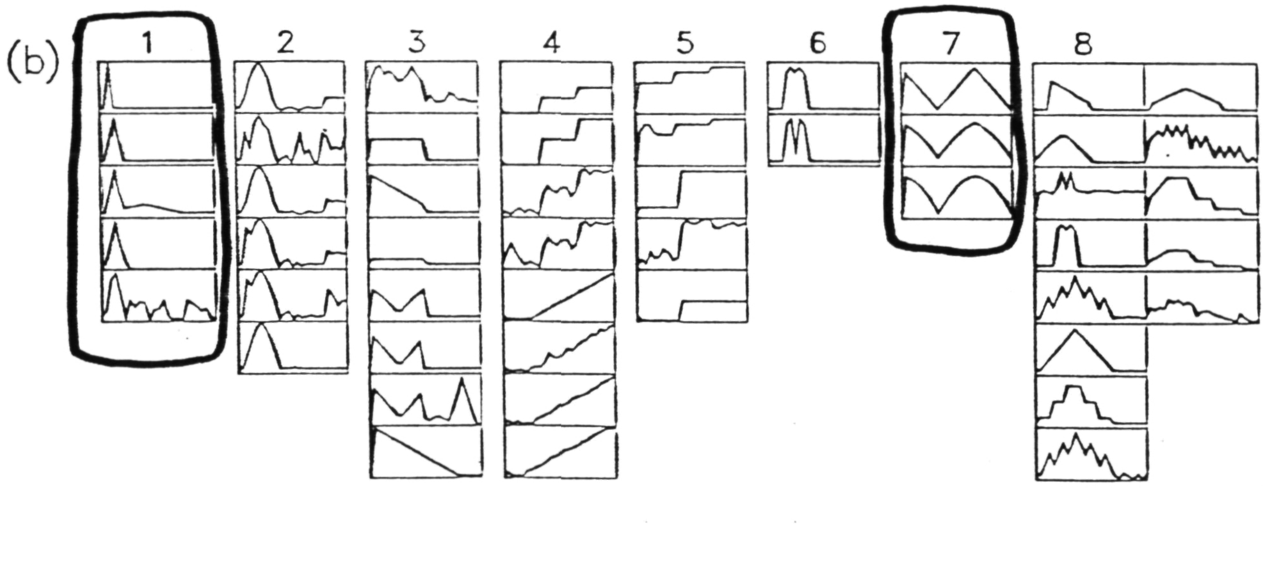


VERY LOW VIGILANCE ⇒ NO SEARCH

FIRST PRESENTATION



THIRD PRESENTATION ONWARDS



USE SIMILARITY INVARIANT

TO CONSTRUCT FRONT-END

FOR SIZE
ROTATION
TRANSLATION

INVARIANT RECOGNITION

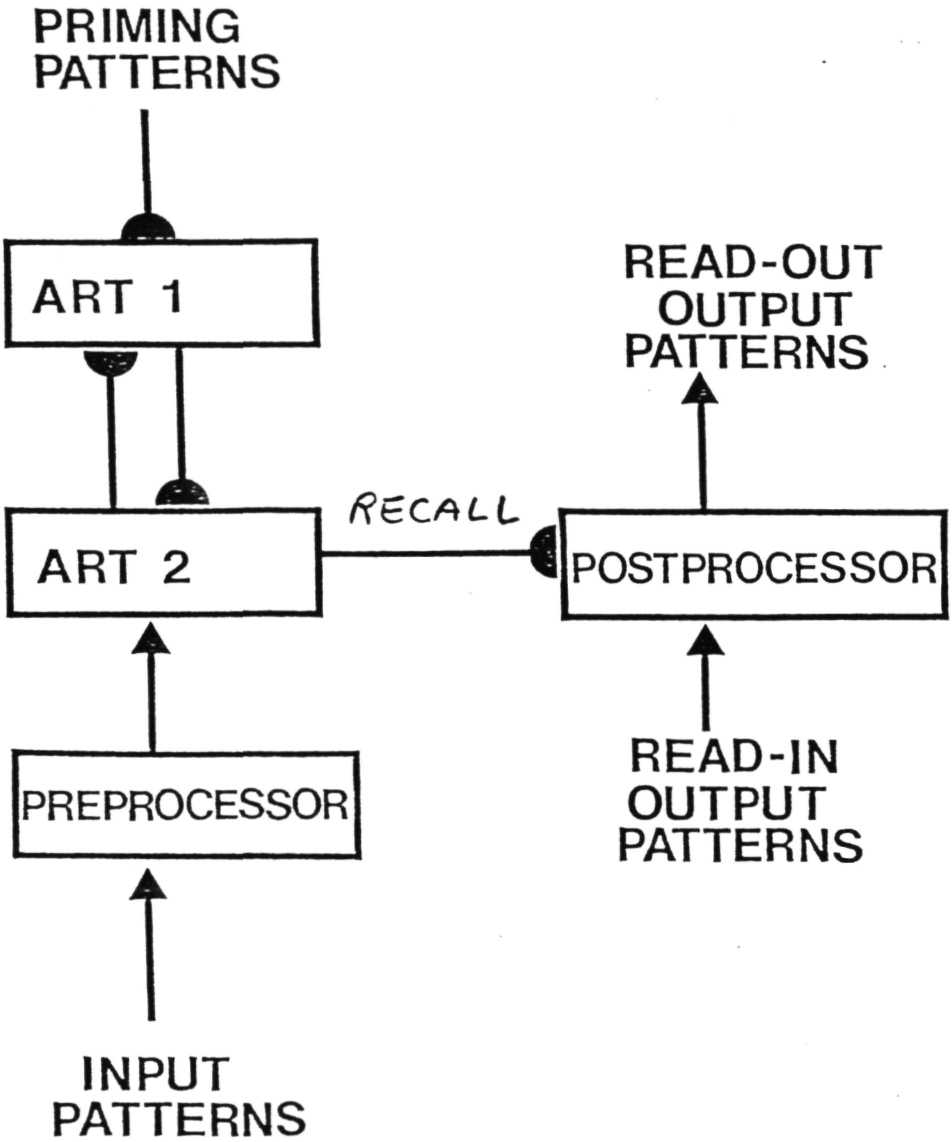
LASER RADAR → BOUNDARY SEGMENTATION

→ FOURIER-MELLIN

→ ART 2

THE 3 R'S:

RECOGNITION
REINFORCEMENT
RECALL



ASSOCIATIVE FAN-IN (INSTARS)	RECOGNITION
ASSOCIATIVE FAN-OUT (OUTSTARS)	RECALL

POSITION INVARIANT

TARGETING

HOW TO COMPUTE THE POSITION OF A TARGET WITH RESPECT TO THE BODY ?

... IN A SELF-ORGANIZING
SELF-CORRECTING
WAY.

(MICHAEL KUPERSTEIN)
1986

POSTERIOR PARIETAL
CORTEX

ACTION (ROBOTICS)

POSITION INVARIANT

HOW TO COMPUTE THE POSITION OF A TARGET?

EYE MOVEMENTS

TARGET POSITION = POSITION OF THE LIGHT ON THE RETINA

PRESENT POSITION = POSITION OF THE EYE IN THE HEAD

TARGET POSITION − PRESENT POSITION

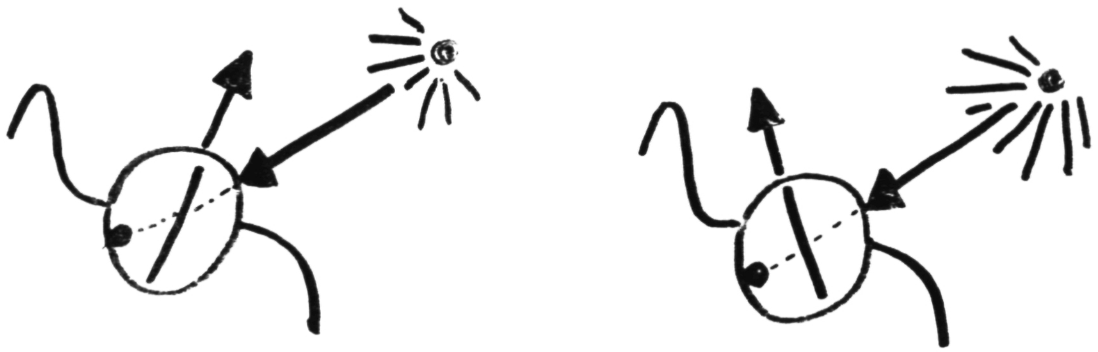
= DIFFERENCE VECTOR

COMPUTES HOW FAR EYE MUST MOVE.

MANY COMBINATIONS OF RETINAL POSITION AND EYE POSITION GENERATE

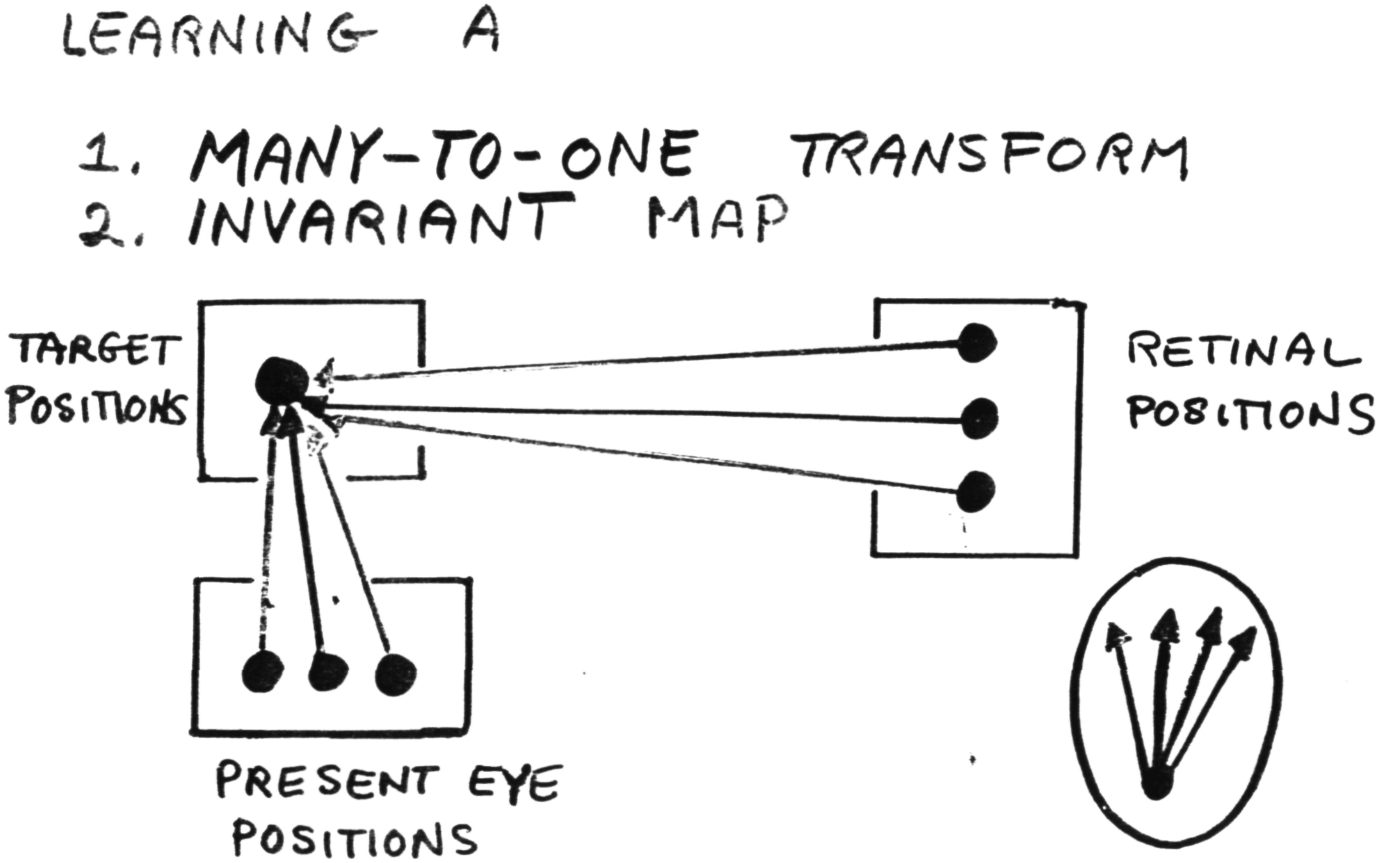
↓

ONE POSITION OF THE LIGHT IN HEAD COORDINATES



LEARNING A

1. MANY-TO-ONE TRANSFORM
2. INVARIANT MAP

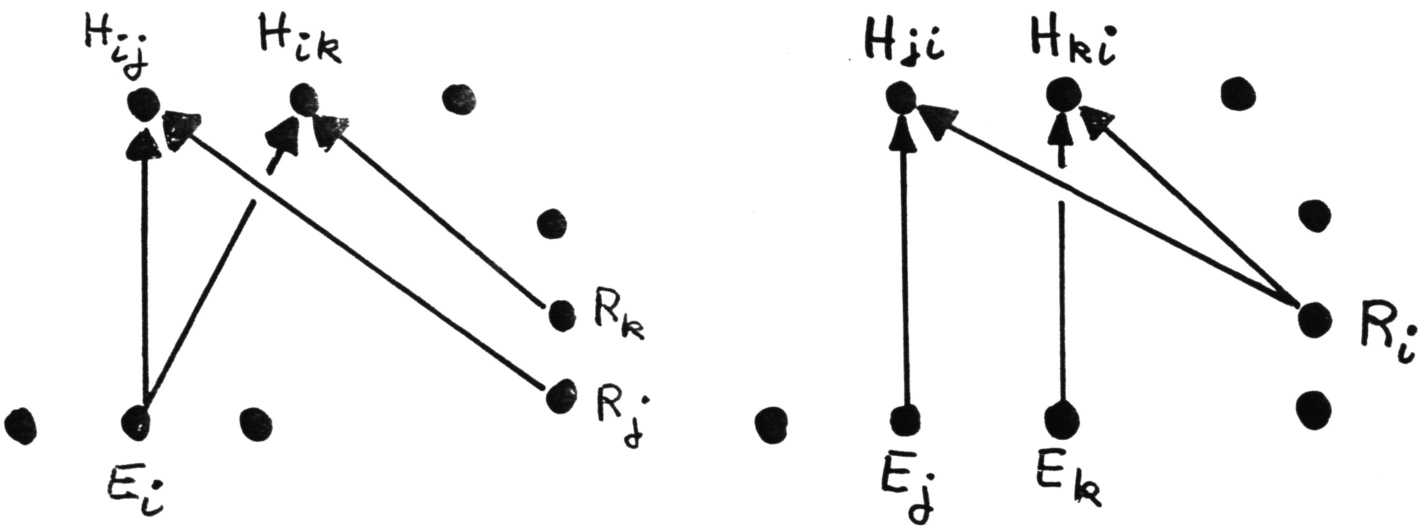


3. SELF-REGULATING AND SELF-CORRECT
(SELF- REPAIRING)

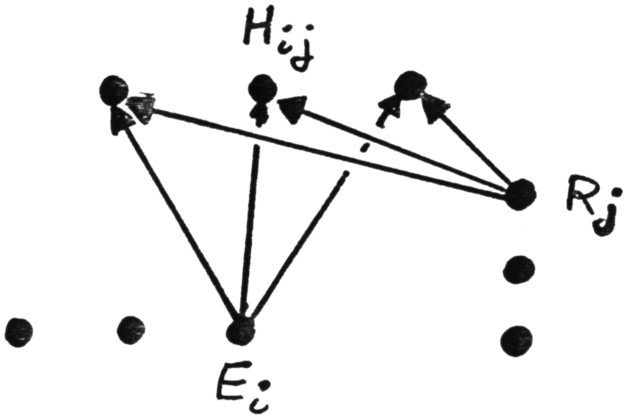
KEY DIFFICULTY (DISTRIBUTED)

EACH R POSITION AND E POSITION CAN ACTIVATE MANY TARGET POSITIONS IN H COORDINATES

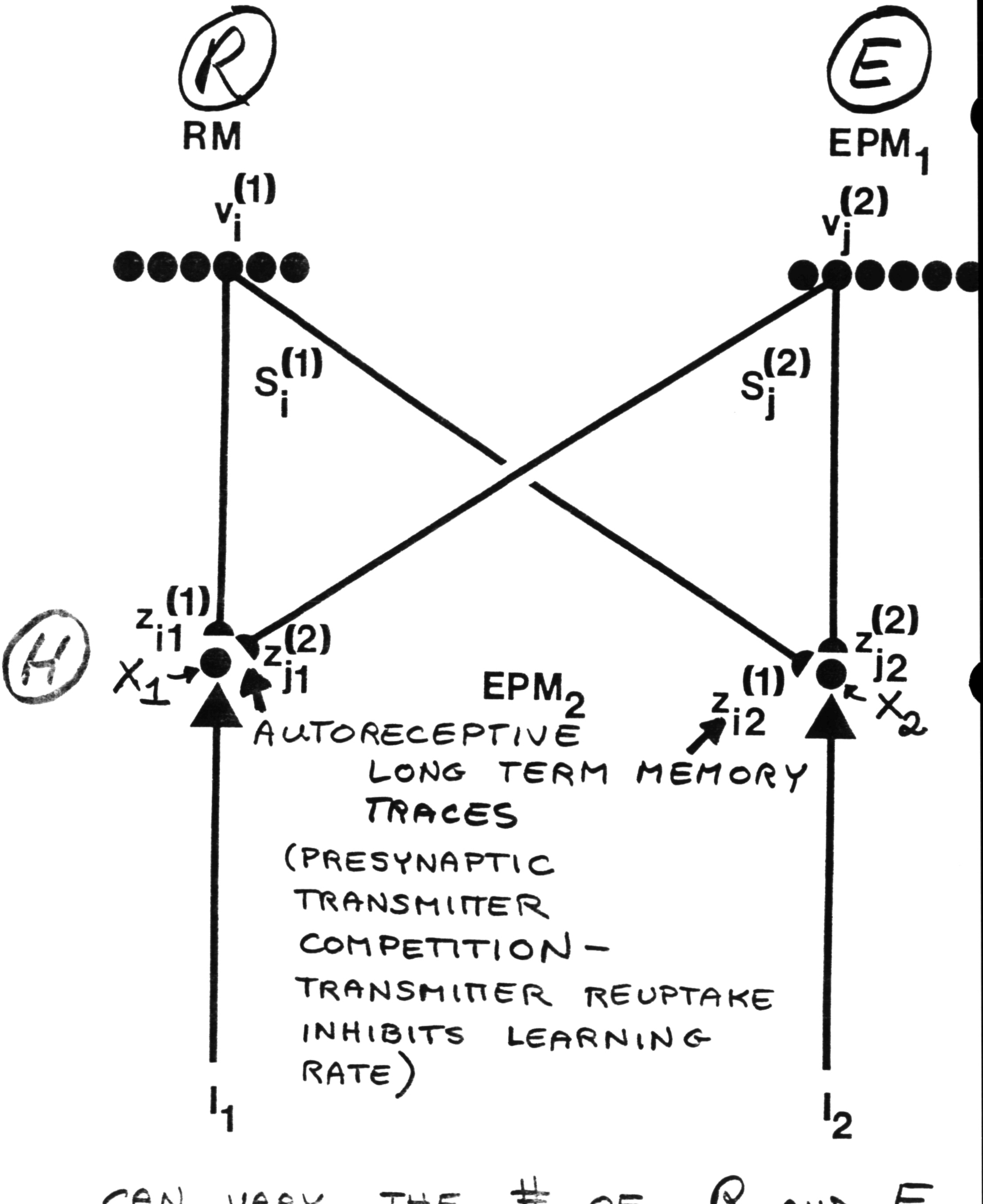
LEARNING



PERFORMANCE



LEARNING A GLOBALLY CONSISTENT RULE BY A DISTRIBUTED NETWORK INTERACTION.



CAN VARY THE # OF R AND E COMBINATIONS

A SPECIALIZED ADDITIVE MODEL!

FAST
SHORT TERM MEMORY:

HEAD-CENTERED PATTERN

$$\frac{d}{dt} x_i = -x_i + I_i + z_i$$

SLOW
LONG TERM MEMORY:

$$z_i = \underbrace{\sum_j S_j^{(1)} z_{ji}^{(1)}}_{\text{EYE}} + \underbrace{\sum_k S_k^{(2)} z_{ki}^{(2)}}_{\text{HEAD}}$$

$$\frac{d}{dt} z_{ji}^{(1)} = \epsilon S_j^{(1)} \left[-A z_{ji}^{(1)} + B x_i - C z_i \right]$$

AUTORECEPTOR TERM

$$\frac{d}{dt} z_{ki}^{(2)} = \epsilon S_k^{(2)} \left[-A z_{ki}^{(2)} + B x_i - C z_i \right]$$

RANDOM INPUTS: $S_j^{(1)}, S_k^{(2)}$.

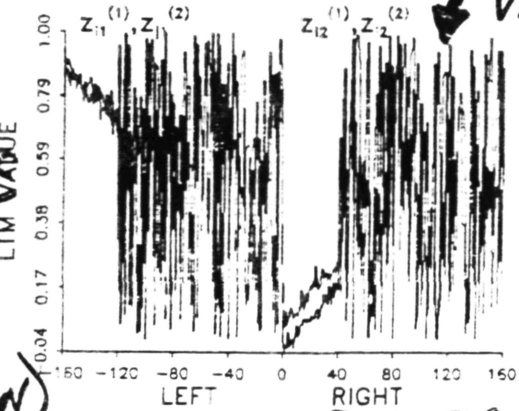
VERY LARGE SINGULAR STOCHASTIC SYSTEM OF ODE's,
TO ADAPTIVELY TRANSFORM VISUAL DATA INTO AN INVARIANT, SELF-REGULATING TARGET POSITION MAP IN HEAD COORDINATES.

INVARIANT SELF-REGULATING LTM MA

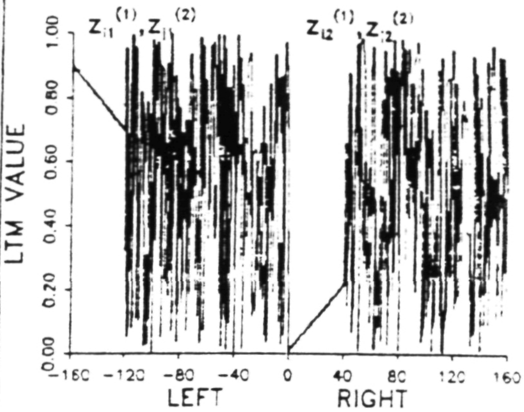
DURING LEARNING | AFTER LEARN

RANDOM INITIAL LTM VALUES

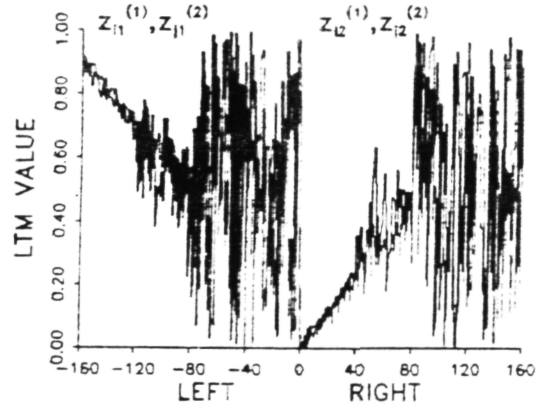
LTM MAPS (RET., AND EYE POSITION)



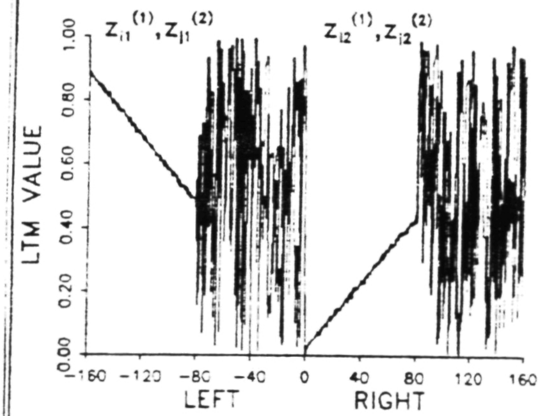
(a) POSITION



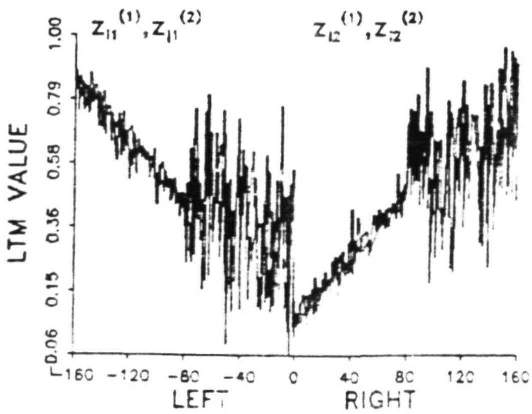
(b)



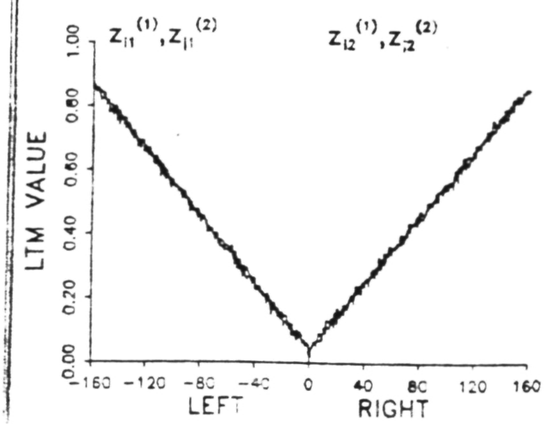
(c) MORE COMBINATIONS



(d)

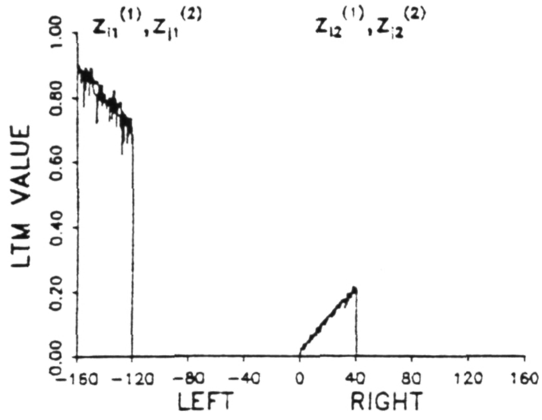


(e) STILL MORE COMBINATIONS

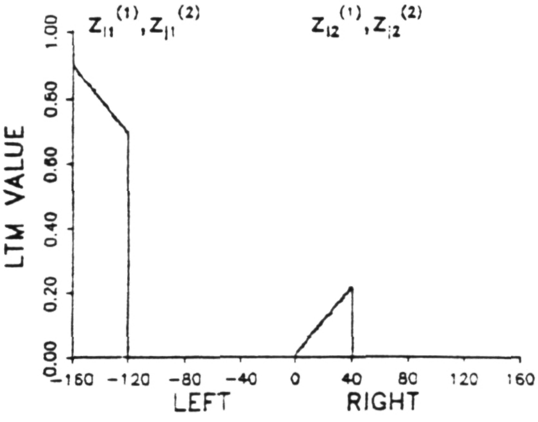


(f)

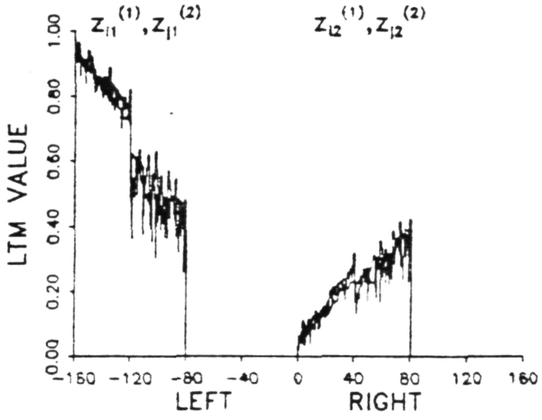
VERY INSENSITIVE TO NOISE!



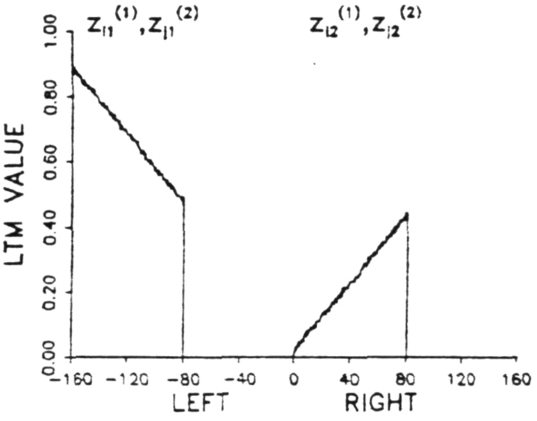
(a)



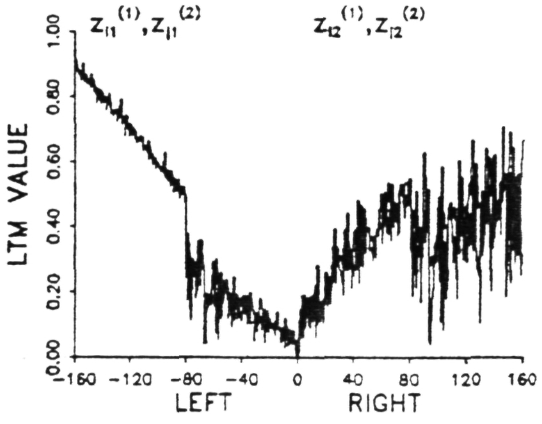
(b)



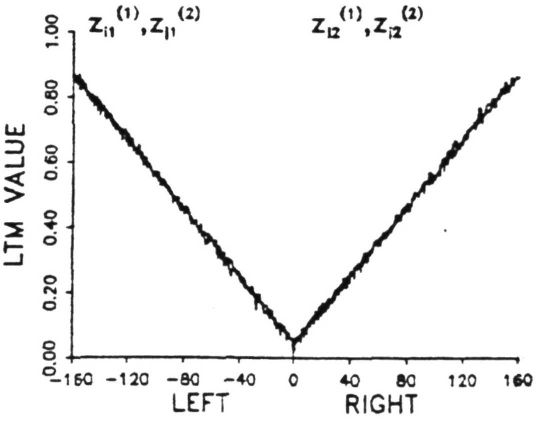
(c)



(d)



(e)



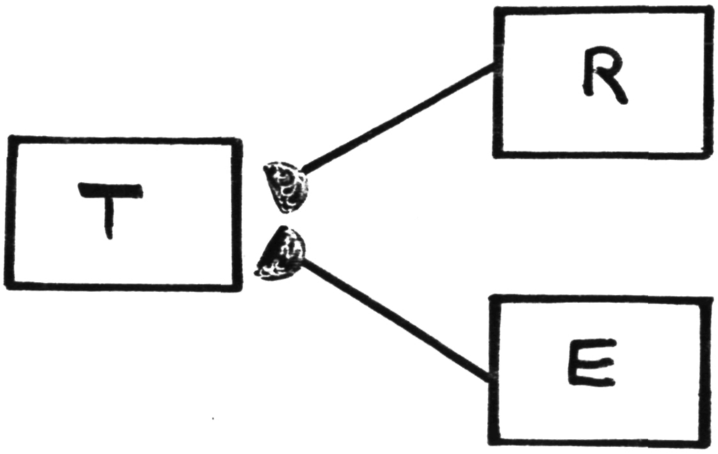
(f)

SOLUTION

NEURAL DYNAMICS OF ADAPTIVE SENSORY-MOTOR CONTROL, 1986
(MICHAEL KUPERSTEIN)

SPECIALIZED ADDITIVE MODEL!

WHERE DO YOU GET GOOD TARGET POSITIONS TO LEARN?!



HOW TO GET STARTED?!
INFINITE REGRESS

VISUALLY REACTIVE MOVEMENT SYSTEM

VISUAL ERROR SIGNALS
MISMATCH

NETWORK
I

VISUAL ERROR SIGNAL

EXTERNAL "TEACHER"

MISMATCH

NETWORK
II

TARGET POSITIONS

INTERNAL "TEACHER

MATCH

POSITION
INVARIANT

TO OVERCOME INFINITE REGRESS:

NETWORK HIERARCHY OF DEVELOPING MOVEMENT SYSTEM

FUNCTION	ANATOMY
1. VISUALLY REACTIVE MOVEMENTS	(SUPERIOR COLLICU[LUS], CEREBELLUM, VISUA[L] CORTEX, ...)
↓	
2. ATTENTIONALLY MODULATED MOVEMENTS	(PARIETAL CORTE[X] ...) TARGET POSITIONS
↓	
3. PLANNED INTENTIONAL MOVEMENT SEQUENCES WHICH DERIVE THEIR ACCURACY FROM (1), BUT CAN IGNORE VISION'S MOMENTARY DEMANDS.	(FRONTAL CORTEX, ...)

MANY FUNCTIONALLY CHARACTERIZED ARCHITECTURES

~ MANY SPECIALIZED BRAIN REGIONS

SYNCHRONY INVARIANT

MULTI-JOINT MOVEMENT

(ARM,
SPEECH ARTICULATORS, ...)

HOW ARE MOVEMENT RATES OF MUSCLES ADJUSTED IN PARALLEL SO DIFFERENT AMOUNTS OF CONTRACTION OCCUR IN EQUAL TIME?

(DANIEL BULLOCK)

PLANNED VS. AUTOMATIC CONTROL

HUMANS

PLAN

1. TARGET POSITION

   WHERE WE WANT TO MOVE

2. SPEED

   HOW FAST TO MOVE

AUTOMATIC

3. PRESENT POSITION

4. UNEXPECTED LOADS AND INERTIAS

5. CHANGES IN MOTOR PLANT

VECTOR INTEGRATION TO ENDPOINT MODEL

(DANIEL BULLOCK)

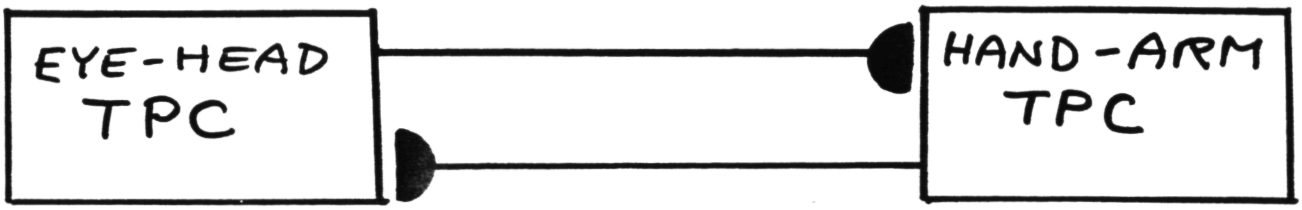
KEY QUESTION:

HOW DO WE LEARN HOW TO REACH WITH AN ARM AN OBJECT THAT WE SEE WITH OUR EYES?

WHAT INFORMATION IS AVAILABLE IN REAL-TIME ON WHICH TO BASE THE LEARNING PROCESS?

PIAGET!

LEARNED ASSOCIATIVE MAP
BETWEEN TARGET POSITION CODES
T P C



CIRCULAR REACTION
(PIAGET)

HOW DOES HAND-ARM TPC

GENERATE A SYNCHRONOUS
TRAJECTORY

NEED AUTOMATIC COMPENSATION
FOR

PRESENT POSITION (PPC)

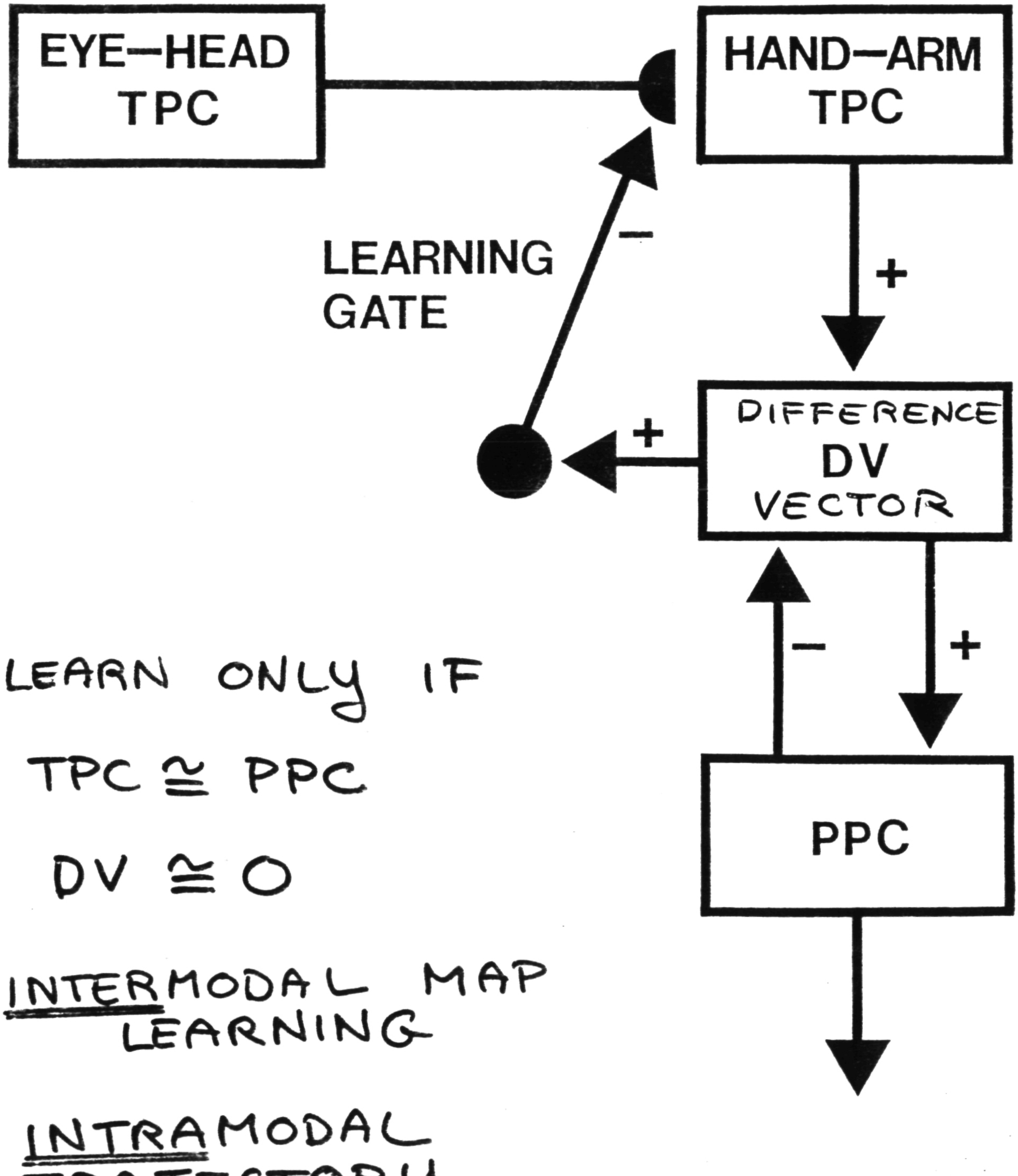
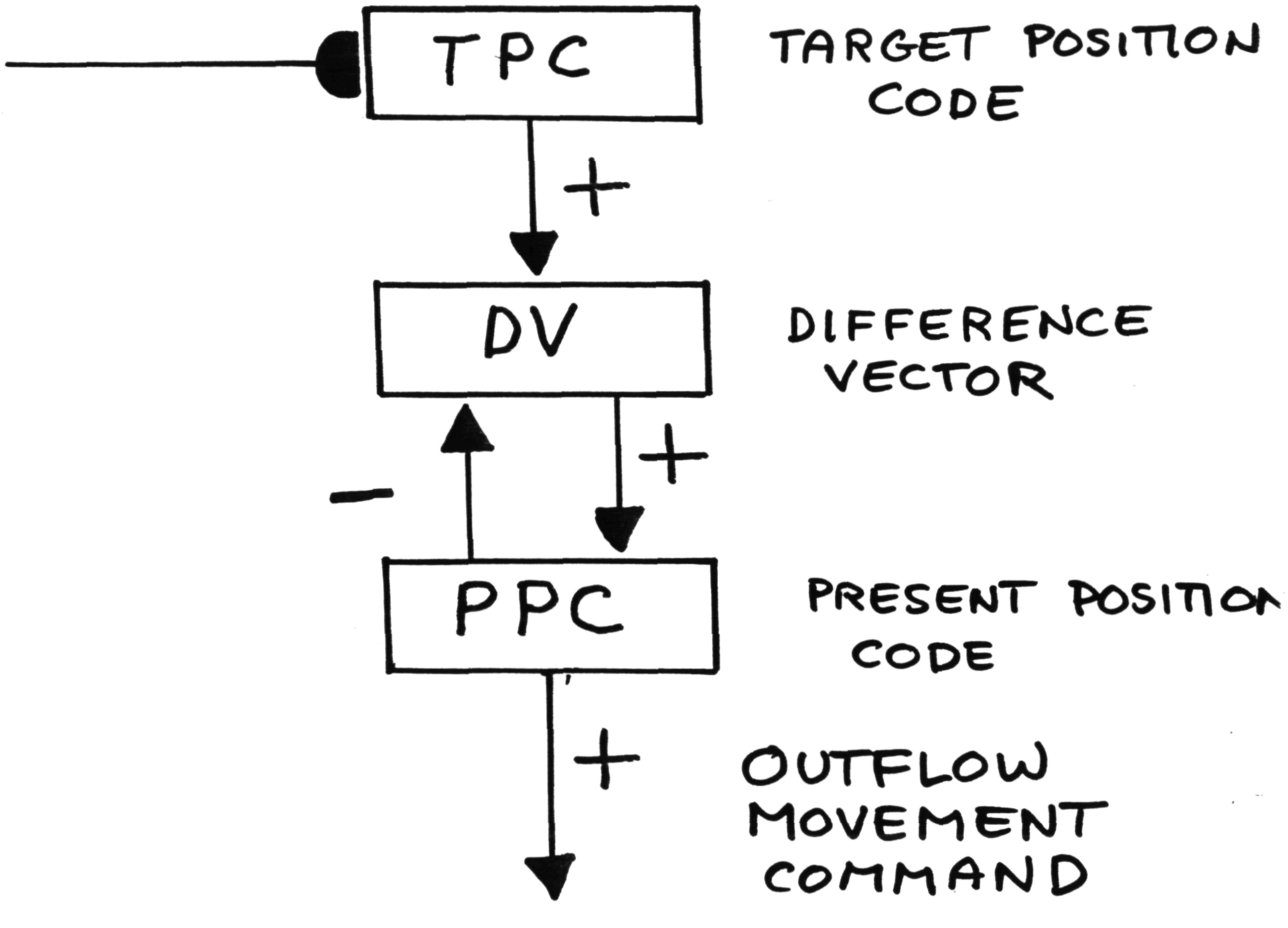

EYE—HEAD TPC
HAND—ARM TPC
LEARNING GATE
−
+
DIFFERENCE DV VECTOR
+
−
+
PPC
LEARN ONLY IF
TPC ≅ PPC
DV ≅ 0
INTERMODAL MAP LEARNING
INTRAMODAL TRAJECTORY FORMATION

Figure 14

VITE MODEL



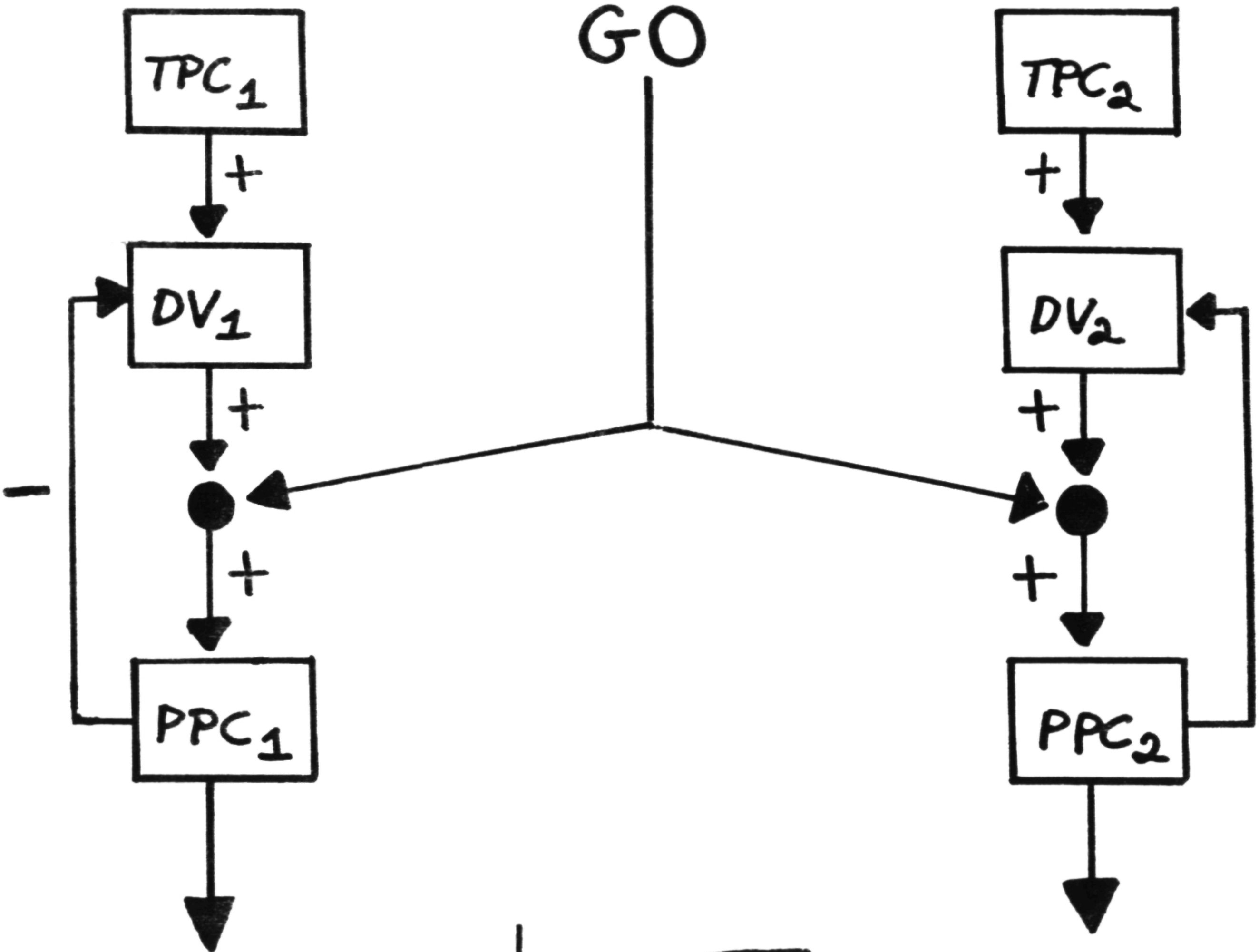
DV IS ADDED TO PPC UNTIL

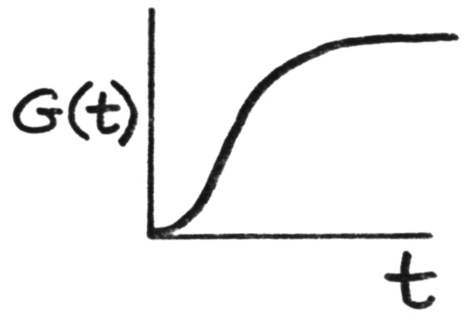
PPC = TPC AND DV = 0.

PPC NOT INFLOW!

HOW IS SPEED CONTROLLED?

NONSPECIFIC, MULTIPLICATIVE GO SIGNAL





GO MECHANISM (VITE)

IS FUNCTIONALLY HOMOLOGOUS TO

ATTENTIONAL GAIN CONTROL (ART)

IS FUNCTIONALLY HOMOLOGOUS TO

AROUSAL SOURCE (AVALANCHE)

⋮

A GENERAL DESIGN PRINCIPLE

FACTORIZATION OF PATTERN + ENERGY

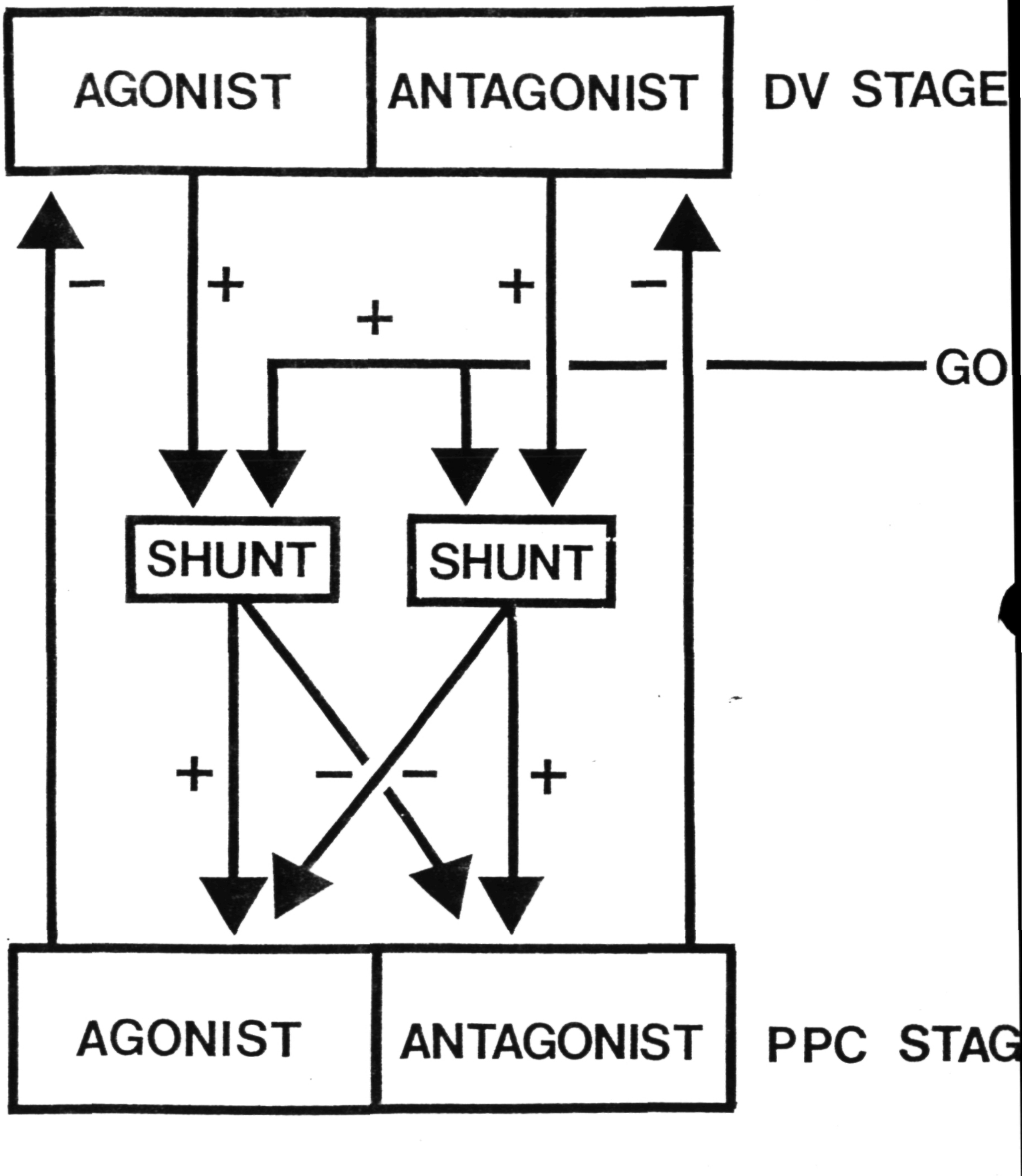
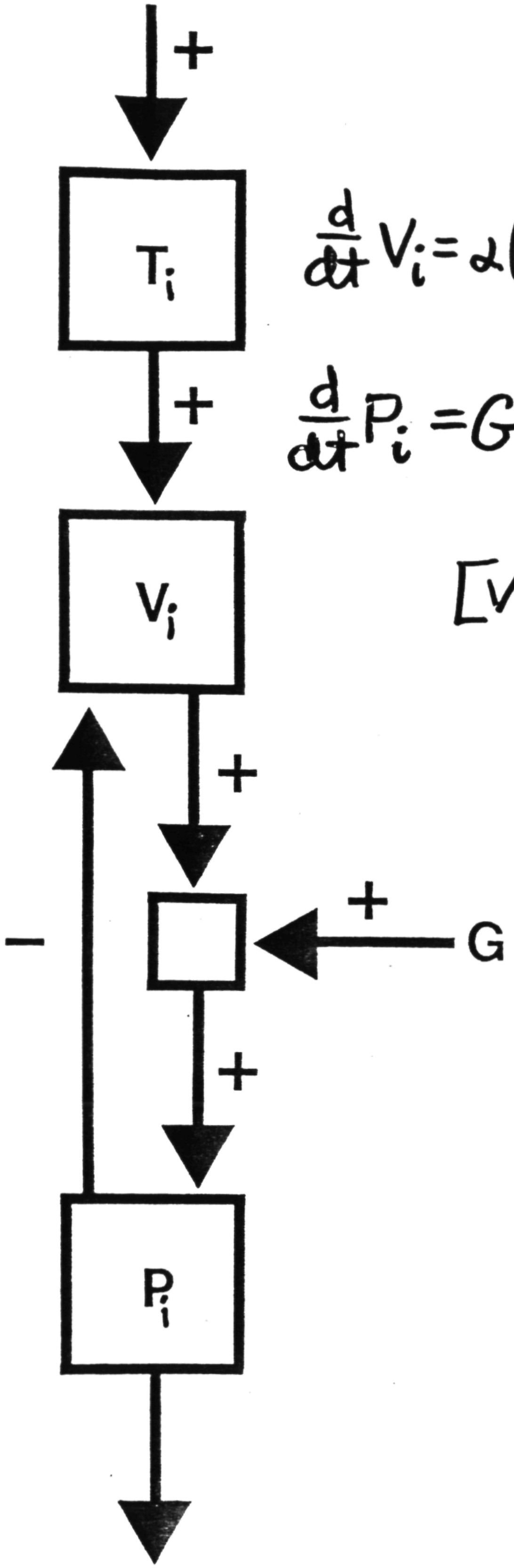

AGONIST
ANTAGONIST
DV STAGE
+
+
+
GO
SHUNT
SHUNT
+
+
AGONIST
ANTAGONIST
PPC STAG

Figure 16



$$\frac{d}{dt}V_i = \alpha(-V_i + T_i - P_i)$$

$$\frac{d}{dt}P_i = G(t)\max(V_i, 0)$$

$$[V_i]^+ = \max(V_i, 0)$$

DIGITAL SWITCH OF TPC
SMOOTH SYNCHRONOUS TRAJECTORY

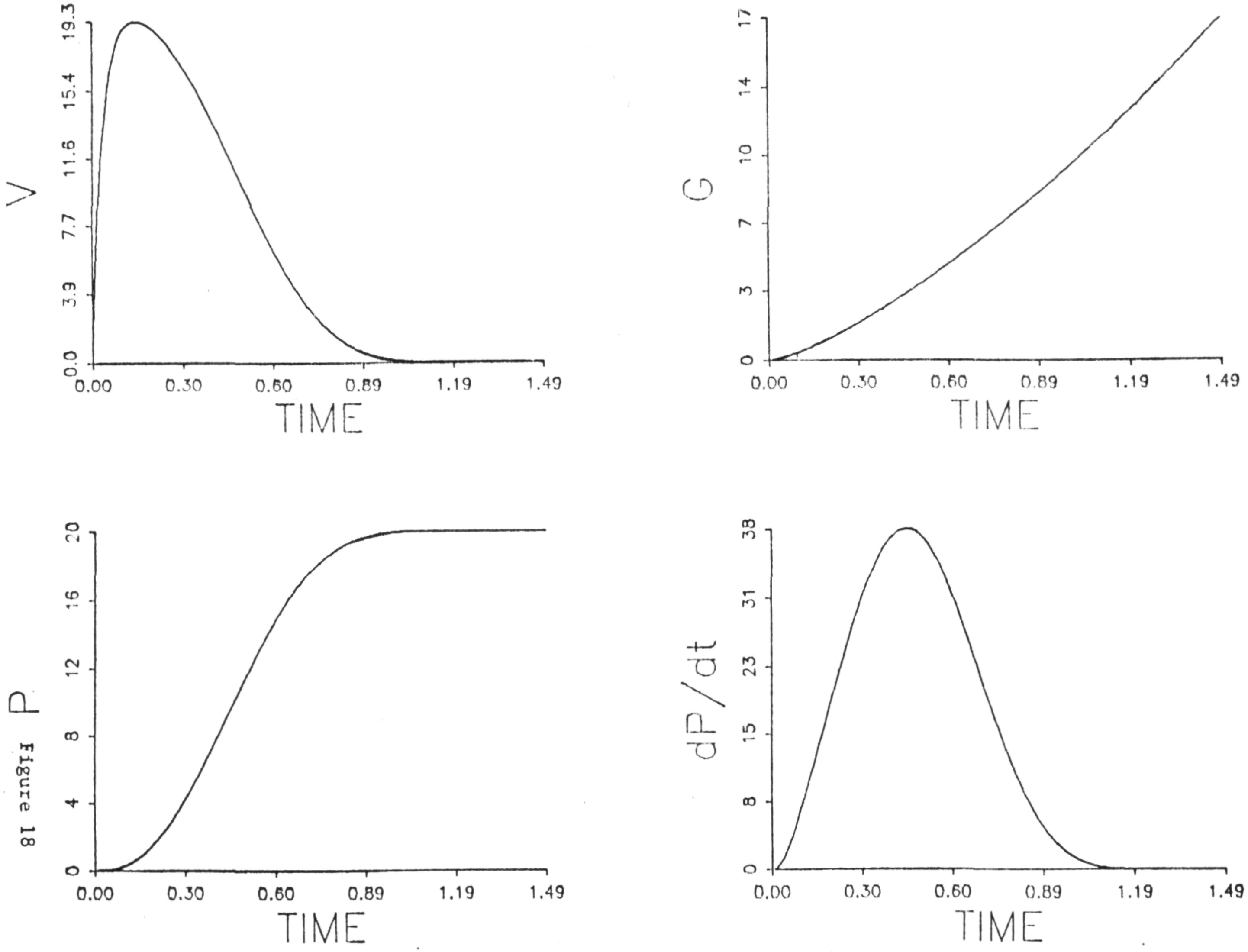
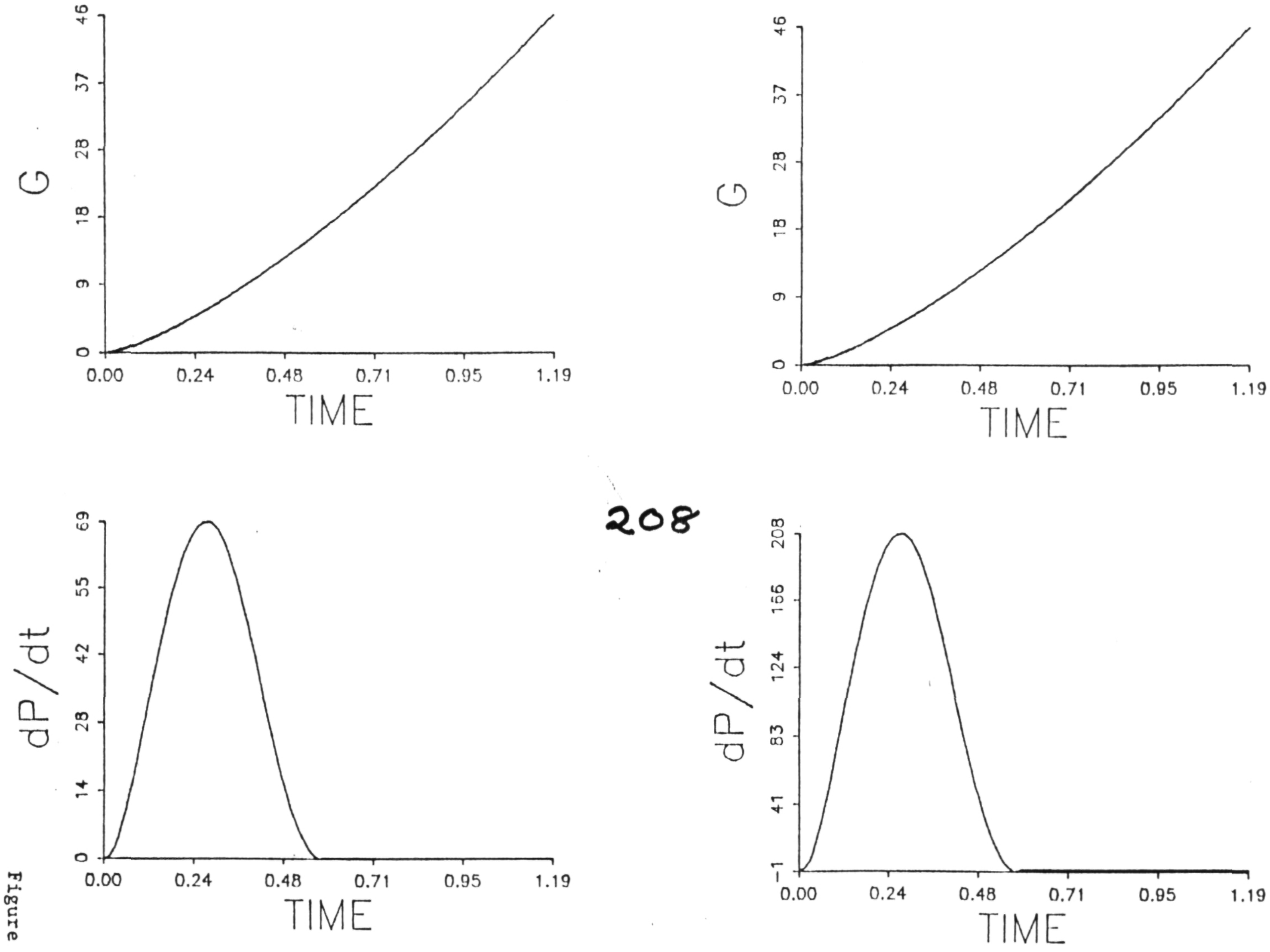


Figure 18

69



(A) (B)

Figure 19

SYNCHRONY AND DURATION INVARIANCE

GIVEN

$$\frac{d}{dt}V = \alpha(-V + T - P)$$

$$\frac{d}{dt}P = G(t)\max(V, 0),$$

WITH $V(0) = 0$.

SWITCH T FROM $T = T_0$ TO

$T = T_1 > T_0$ AT $t = 0$.

PROVE:

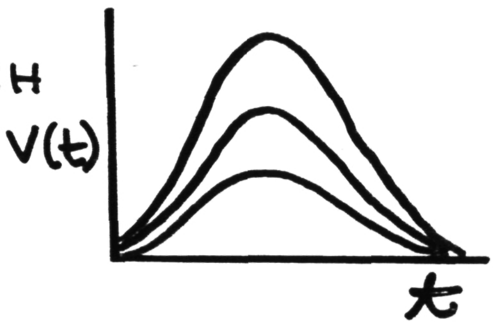
$$P(t) = P(0) + (T_1 - T_0)\,g(t)$$

WHERE $g(t)$ IS INDEPENDENT OF T_0 AND T_1.

EXPLAINS LARGE PARAMETRIC DATA BASE

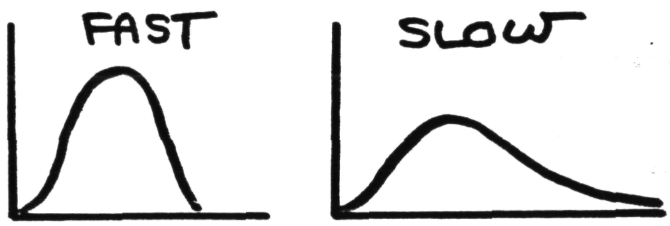
VELOCITY PROFILES FROM MOVEMENTS OF SIMILAR DURATION BUT UNEQUAL DISTANCE SUPERIMPOSE AFTER VELOCITY AXIS RESCALING

FREUND AND BÜDINGEN
ATKESON AND HOLLERBACH



VELOCITY PROFILE ASYMMETRY VARIES WITH DURATION

BEGGS AND HOWARTH
ZELAZNIK *et al*



(EVIDENCE AGAINST HOGAN'S MINIMUM JERK MODEL)

PRESENT POSITION COMMANDS ARE GRADUALLY UPDATED

BIZZI *et al*

(EVIDENCE AGAINST "SPRING-TO-ENDPOIN[T]" MODELS)

LOGARITHMIC SPEED-ACCURACY TRADEOFF (FITTS' LAW)

$$MT = a + b \log_2 \left(\frac{2D}{W} \right)$$

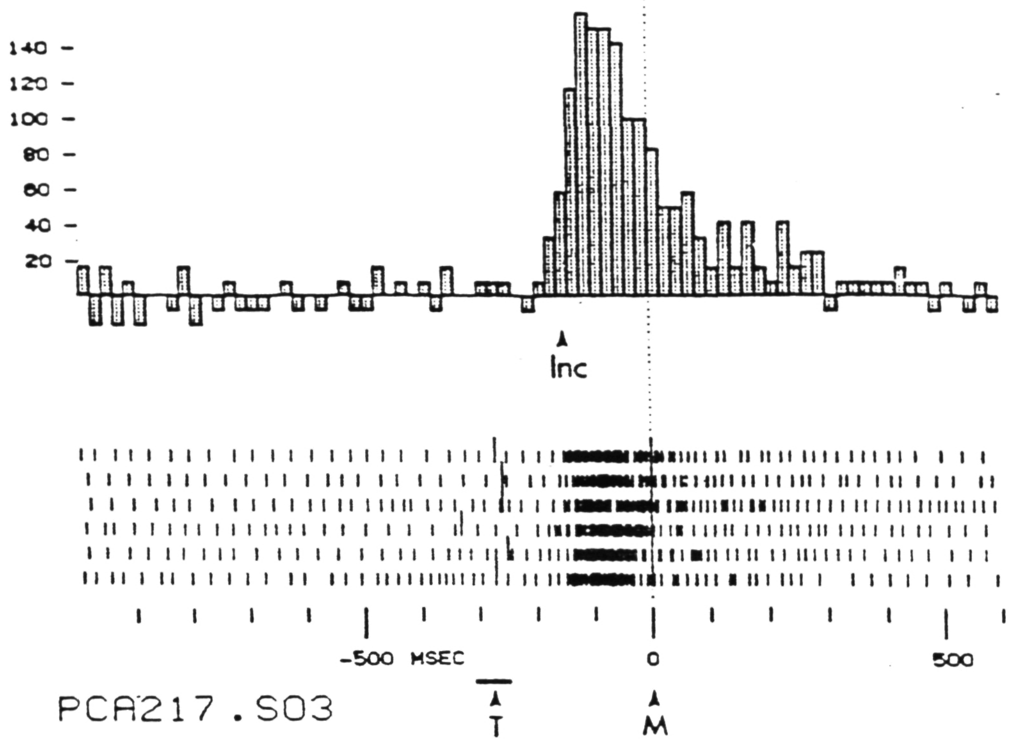
MT = MOVEMENT TIME

D = DISTANCE MOVED

W = TARGET WIDTH

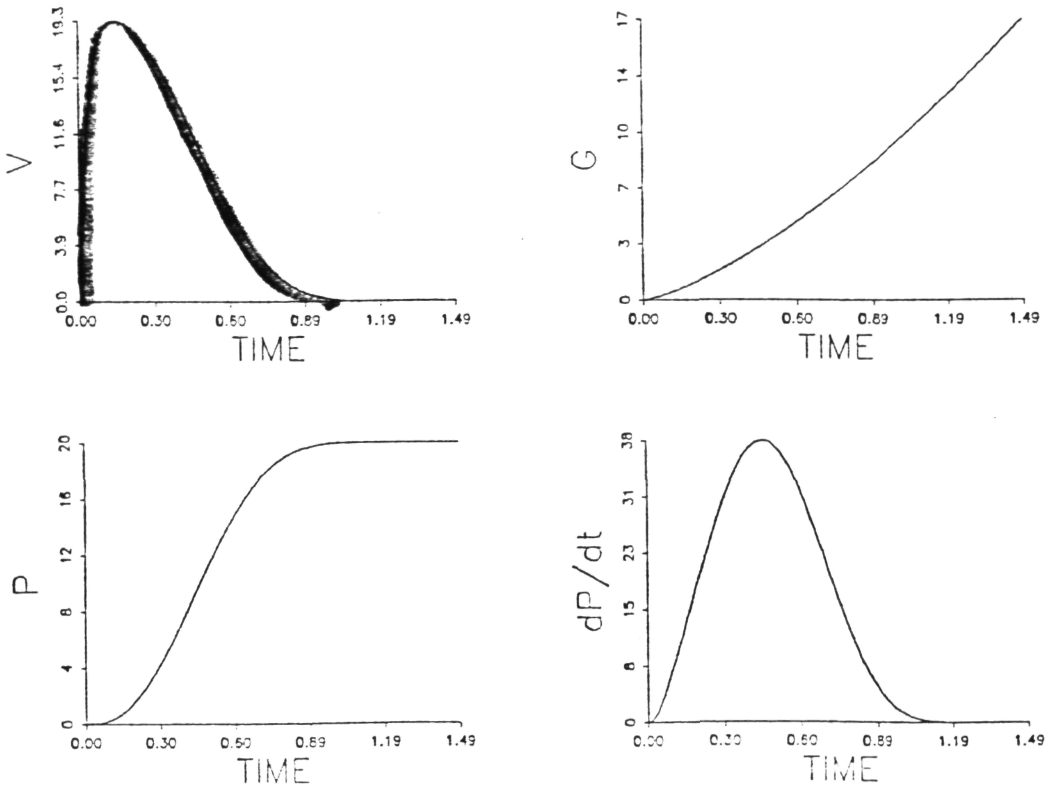
VECTOR CELL IN PREFRONTAL CORTEX

SPIKES/SEC (DEV. FROM MEAN)



(GEORGOPOULOS et al; EVARTS AND TANJI)

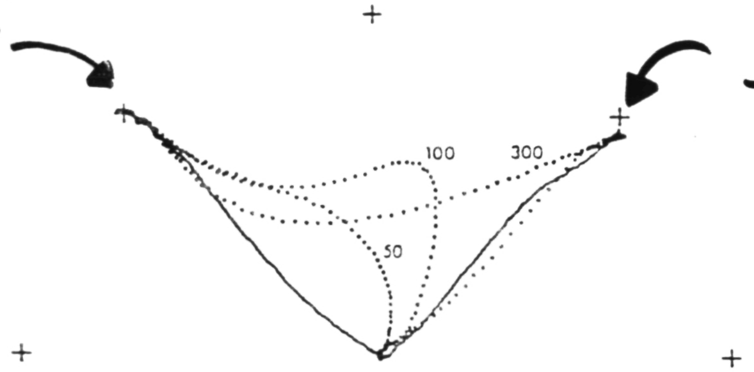
DV NODE IN VITE MODEL



AN ADAPTIVE EMERGENT PROPERT

SWITCHED TARGET

INITIAL TARGET

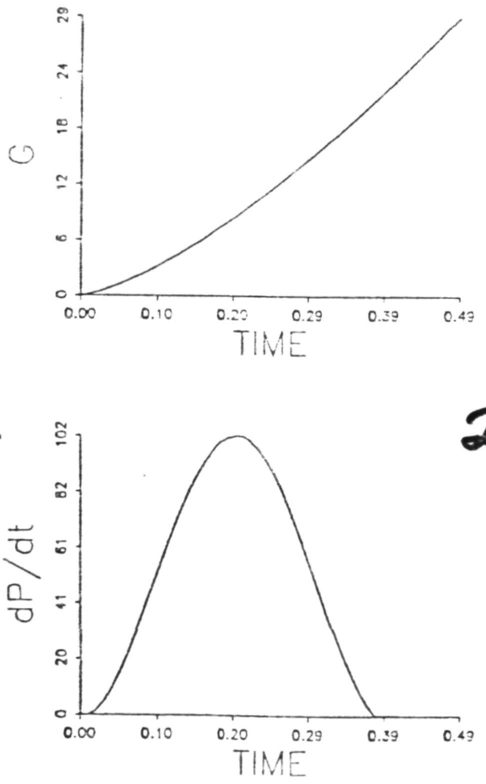


SEAMLESS TRAJECTORY CORRECTION AND
"ANOMALOUS" MULTIPLICATION OF PEAK VELOCITY
(GEORGOPOULOS <u>et</u> <u>al</u>)

LOCAL VIOLATION OF FITTS' LAW

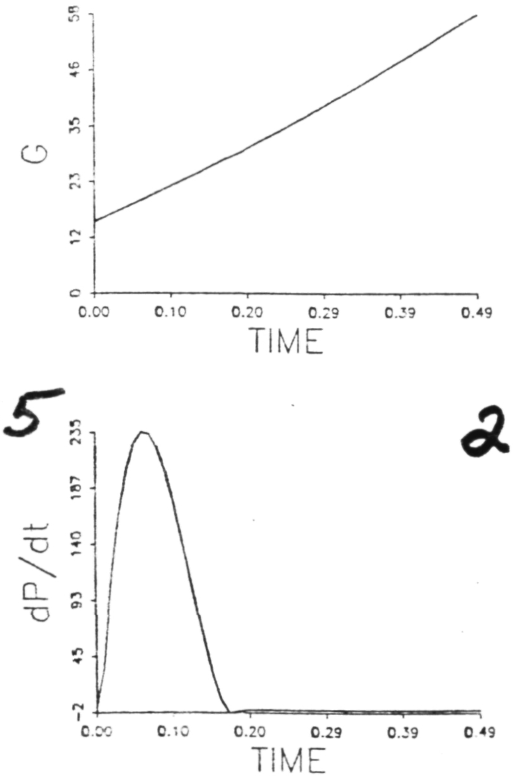
GO

102



(A)

235



(B)

2.3X

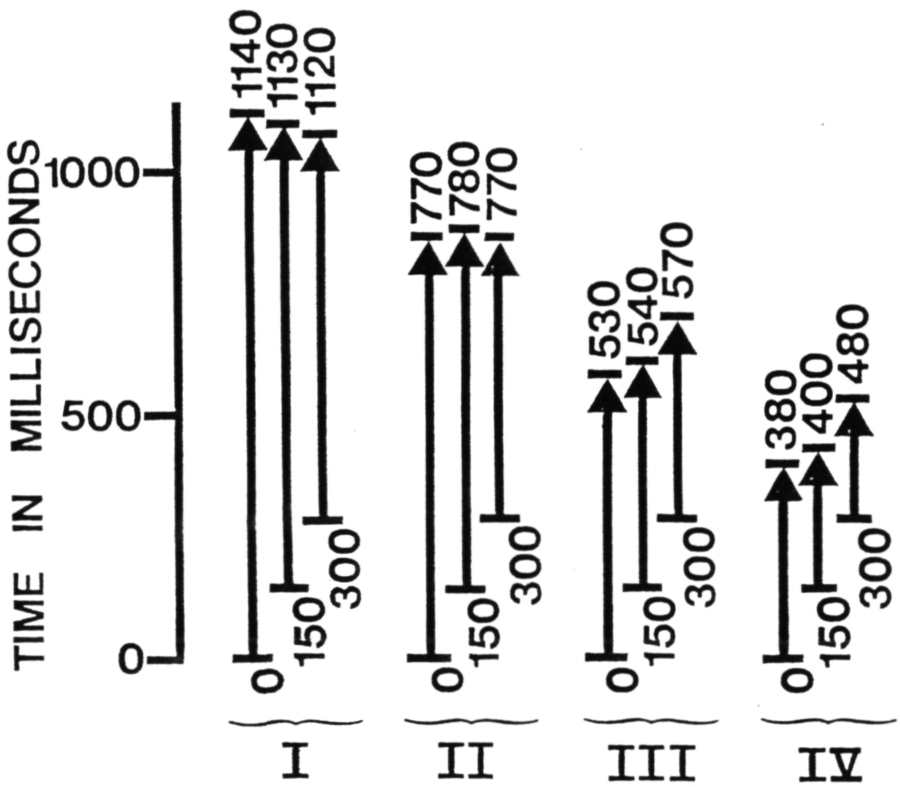
EQUIFINALITY

AUTOMATIC COMPENSATION FOR STAGGERED ONSET TIMES

⊥ SYNERGIST BEGINS CONTRACTION

⊤ SYNERGIST ENDS CONTRACTION

FAULT TOLERANT



PRESENT POSITION

PASSIVE VS. ACTIVE MOVEME

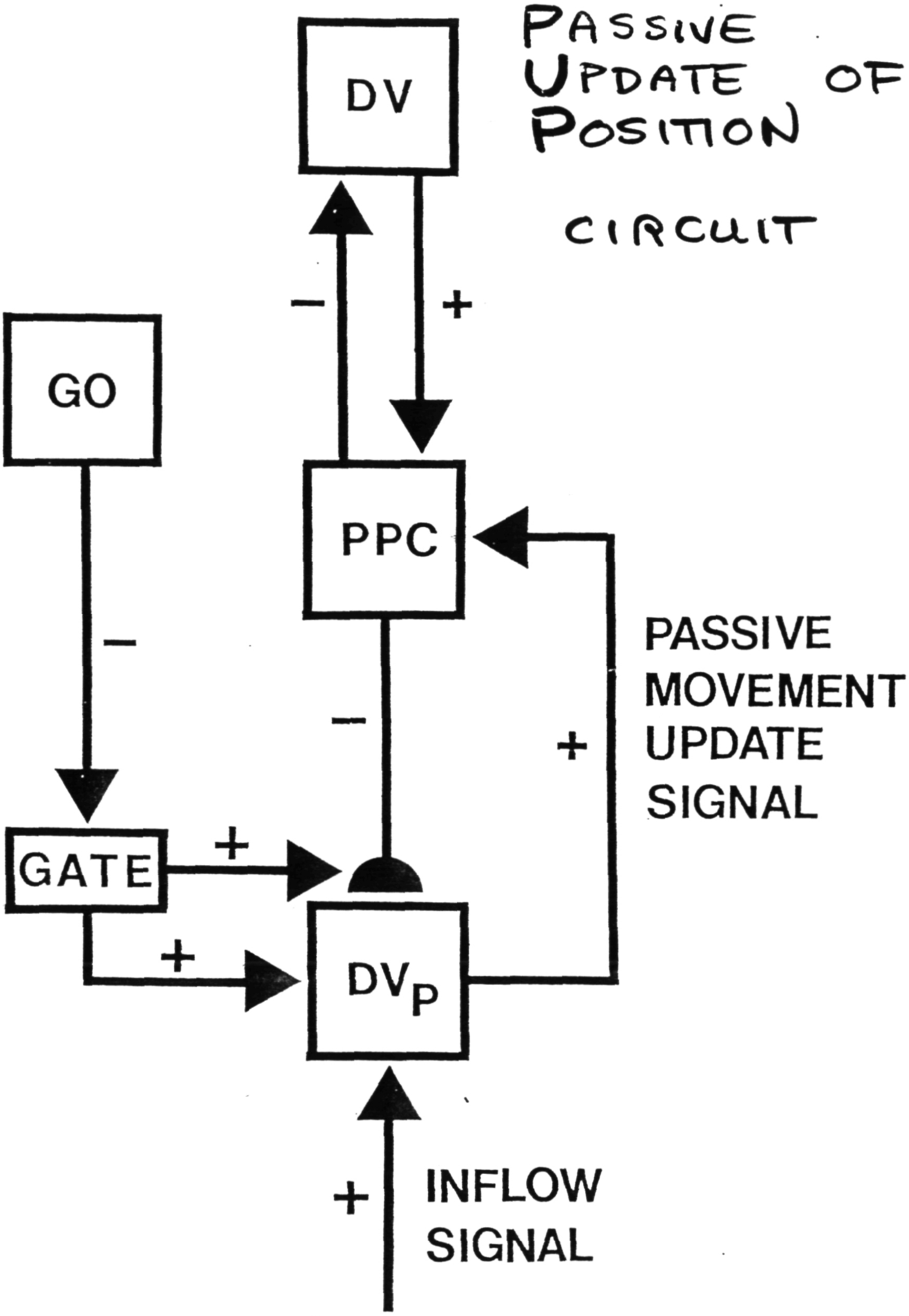
GRAVITY
EXTERNAL FORCES

OUTFLOW VS. INFLOW

ADAPTIVE COORDINATE CHANGE

PASSIVE UPDATE OF POSITION

CIRCUIT



ADAPTIVE VECTOR ENCODER
ADAPTIVE COORDINATE CHANGE

VITE + PUP

VECTOR

$$\frac{d}{dt} V_i = \alpha(-V_i + T_i - P_i)$$

PPC

$$\frac{d}{dt} P_i = G[V_i]^+ + G_p[M_i]^+$$

$$G G_p = 0$$

OUTFLOW-INFLOW MATCH

$$\frac{d}{dt} M_i = -\beta M_i + \gamma_i I_i - z_i P_i$$

ADAPTIVE GAIN CONTROL (LTM

$$\frac{d}{dt} z_i = \delta G_p(-\epsilon z_i + [M_i]^+).$$

A MULTI-ARCHITECTURAL ADAPTIVE SYSTEM

MULTIPLE LEARNING PROBLEMS $\sim$ MULTIPLE CIRCUITS

SUMMARY

EMERGENT INVARIANT	BEHAVIOR
SIMILARITY	RECOGNITION
POSITION	TARGETING
SYNCHRONY	MULTI-JOINT MOVEMENT

SPECIALIZED INFORMATION PROCESSING

SPECIALIZED LEARNING LAWS

REAL-TIME NONLINEAR FEEDBACK

ALL EXAMPLES OF A FEW GENERAL EQUATIONS WITH SPECIALIZED PARAMETER CHOICES

U.S. GOVERNMENT PRINTING OFFICE 1988 561-009/

NOTES

OMIT

Bart Kosko, Ph.D.

University of
Southern California
Los Angeles, California

Dr. Kosko holds bachelor degrees in economics and philosophy from the University of Southern California (USC), a masters degree in mathematics from the University of California at San Diego, and a Ph.D. degree in electrical engineering from the University of California at Irvine. He is an assistant professor in the electrical engineering department at USC and a member of the Signal and Image Processing Institute. Dr. Kosko is also program chairman of the 1988 IEEE International Conference on Neural Networks (ICNN-88); organizing chairman and program chairman of ICNN-87; and is an associate editor, as well as technology news editor, of the journal Neural Networks.

FUZZY THEORY AND NEURAL NETWORKS

Abstract

Does fuzziness differ from probability? How does fuzzy theory relate to neural networks? These questions are addressed. There are many other connections between neural networks and fuzzy theory. Besides fuzzy entropy minimizers, fuzzy associative memories (FAM's) map fuzzy subsets to fuzzy subsets. Simple FAM's can be constructed using a fuzzy Hebb law and max./min. composition instead of vector-matrix multiplication. Another example is fuzzy causal networks, or fuzzy-cognitive-maps, feedback-knowledge networks that admit degrees of causality and perform forward-chaining inference without graph search.

NOTES

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

S2-61
ABS. ONLY

N91-71352
15308

L. A. Zadeh, Ph.D.

University of California
Berkeley, California

Dr. Zadeh received his B.S.E.E. from the University of Teheran, Iran, in 1942; S.M.E.E. from the Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, in 1946; and Ph.D. from Columbia University, New York, New York, in 1949. He has been a professor of electrical engineering and computer sciences at the University of California, Berkeley, California, since 1959. Dr. Zadeh was an instructor in electrical engineering at Columbia University from 1946 to 1950, assistant professor from 1950 to 1953, associate professor from 1953 to 1957, and professor from 1957 to 1959. He has been a member of the Institute for Advanced Study, Princeton, New Jersey, since 1956; was a visiting professor of electrical engineering at MIT in 1962 and 1968; was a visiting scientist at the IBM Research Laboratory in San Jose, California, in 1968, 1973, and 1977; and was a visiting scholar at the Artificial Intelligence Center, SRI International, in Menlo Park, California, in 1981.

DISPOSITIONAL LOGIC AND COMMONSENSE REASONING

Abstract

Dispositional logic (DL) is a branch of fuzzy logic which is concerned with inference from dispositions, or propositions, which are preponderantly, but not necessarily, true. Simple examples of dispositions are: birds can fly, snow is white, and Swedes are blonde. The importance of the concept of a disposition derives from the fact that much of commonsense knowledge may be viewed as a collection of dispositions. Dispositional logic provides an alternative approach to the theories of default reasoning, nonmonotonic reasoning, circumscription, and other widely-used approaches to commonsense reasoning. The premises in DL are assumed to be of the form usually (X is A) or usually (Y is B if X is A), where A and B are fuzzy predicates which play the role of elastic constraints on the variables X and Y. Inference from such premises reduces, in general, to the solution of a nonlinear program. In many cases, an inference rule in DL has the form of a fuzzy syllogism. The importance of dispositional logic transcends its function as a basis for formalization of commonsense reasoning. Viewed in a broader perspective, it underlies the remarkable human ability to make rational decisions in an environment of uncertainty and imprecision.

NOTES

53

53-60

ABS. ONLY

N91-71353

Takeshi Yamakawa, Ph.D.

Kumamoto University
Kumamoto, Japan

Dr. Yamakawa received his B.E. degree in electronics engineering from the Kyushu Institute of Technology in 1969, and his M.E. and Ph.D. degrees in electrochemistry from Tohoku University in 1971 and 1974, respectively. In 1974, Dr. Yamakawa joined the faculty of engineering at Tohoku University as a research assistant in electronics. In 1977, he joined the faculty of engineering at Kumamoto University as a research assistant in electrical engineering. Since 1981, he has been an associate professor of electrical engineering and consumer sciences at Kumamoto University. Dr. Yamakawa is a member of the editorial board of the International Journal of Fuzzy Sets and Systems and a referee for the computer society of IEEE. He is also a program committee chairman of the international workshop to be held at Iizuka, Fukuoka, Japan, from August 20 to 24, 1988, and a program committee member of three international symposiums/conferences. Dr. Yamakawa's research interests include development of fuzzy logic hardware systems and multiple-valued logic circuits (both in integrated circuits) as well as biomedical sensors. He is a pioneer of fuzzy chips and fuzzy computer hardware, and has applied for 53 patents in Japan, 7 patents in the United States, and 5 patents in Europe.

A FUZZY MICROPROCESSOR - A NOVEL DEVICE FOR HIGH-SPEED APPROXIMATE REASONING

Abstract

A fuzzy controller hardware device demonstrated in the second IFSA Congress has distinctive features: (1) high speed (1 000 000 FIPS); (2) easy programming; (3) suitability for nonlinear and/or time-variant systems; and (4) robustness against the noise, temperature change, power supply fluctuation, and defect of transistors. The hardware also has a slight misprogramming. The rule board and the defuzzifier board are reduced to small chips. They are a rule chip and a defuzzifier chip. By employing these two types of chips, a sophisticated fuzzy controller hardware system can be easily implemented.

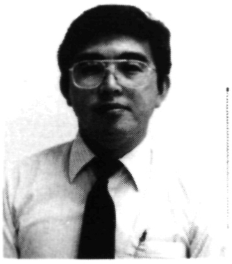
NOTES

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

54

S4-61
15305

N91-71354 P.15



Masaki Togai, Ph.D.

Togai InfraLogic, Inc.
Westlake Village,
California

Dr. Togai received his M.S. and Ph.D. degrees in electrical engineering in 1977 and 1982, respectively, from Duke University. He is president and chief executive officer of Tagai InfraLogic, Inc. Dr. Togai has spent the last 10 years leading fuzzy logic development groups at Duke University, AT&T Bell Laboratories, and Rockwell International. He is best known in the industry for developing the world's first fuzzy microchip for real-time approximate reasoning. He is a member of the board of directors of the North American Fuzzy Information Processing Society (NAFIPS), a member of the American Association of Artificial Intelligence, IEEE, International Fuzzy Systems Association (IFSA), and Sigma xi. In addition, Dr. Togai is the editor-in-chief of the Japan Artificial Intelligence Newsletter; and is an associate editor for the Information Sciences, and the Journal of Approximate Reasoning. He is an author of two books: "Intelligent Robotic Systems", and "Approximate Reasoning in Expert Systems". He has authored and coauthored more than 30 papers.

FUZZY AND NEURAL NET PROCESSOR AND ITS PROGRAMMING ENVIRONMENT

Abstract

The fuzzy logic inference processor (FLIP) is a slave processor designed to speed rule evaluation in high-speed, real-time oriented expert systems. It interfaces easily as a slave processor to standard microprocessors and microcontrollers, and is capable of operating without intervention from the host system. The FLIP device is capable of inferencing using two distinct paradigms: fuzzy and neural. The fuzzy paradigm grades the observation values as to their degree of support of the premise, then weighs and merges conclusions based upon the degree of support each premise receives. The neural paradigm weighs each of the inputs, sums all of the weighted inputs, then applies a transfer function to derive the output. Any combination of these paradigms may be included in a knowledge base. The software system to support the development of fuzzy logic system or neural net descriptions for the FLIP is also under development. This user friendly software interfaces FLIP for evaluation of fuzzy and neural systems, allowing considerable flexibility in developing rules and rule evaluations with capacity for trace and truth maintenance. Use of symbolic representation and "human definitions" greatly simplifies the job of knowledge acquisition.

SOFTWARE ENVIRONMENT

Software developed in ANSI Standard C

Graphical interface developed in Microsoft Windows ™

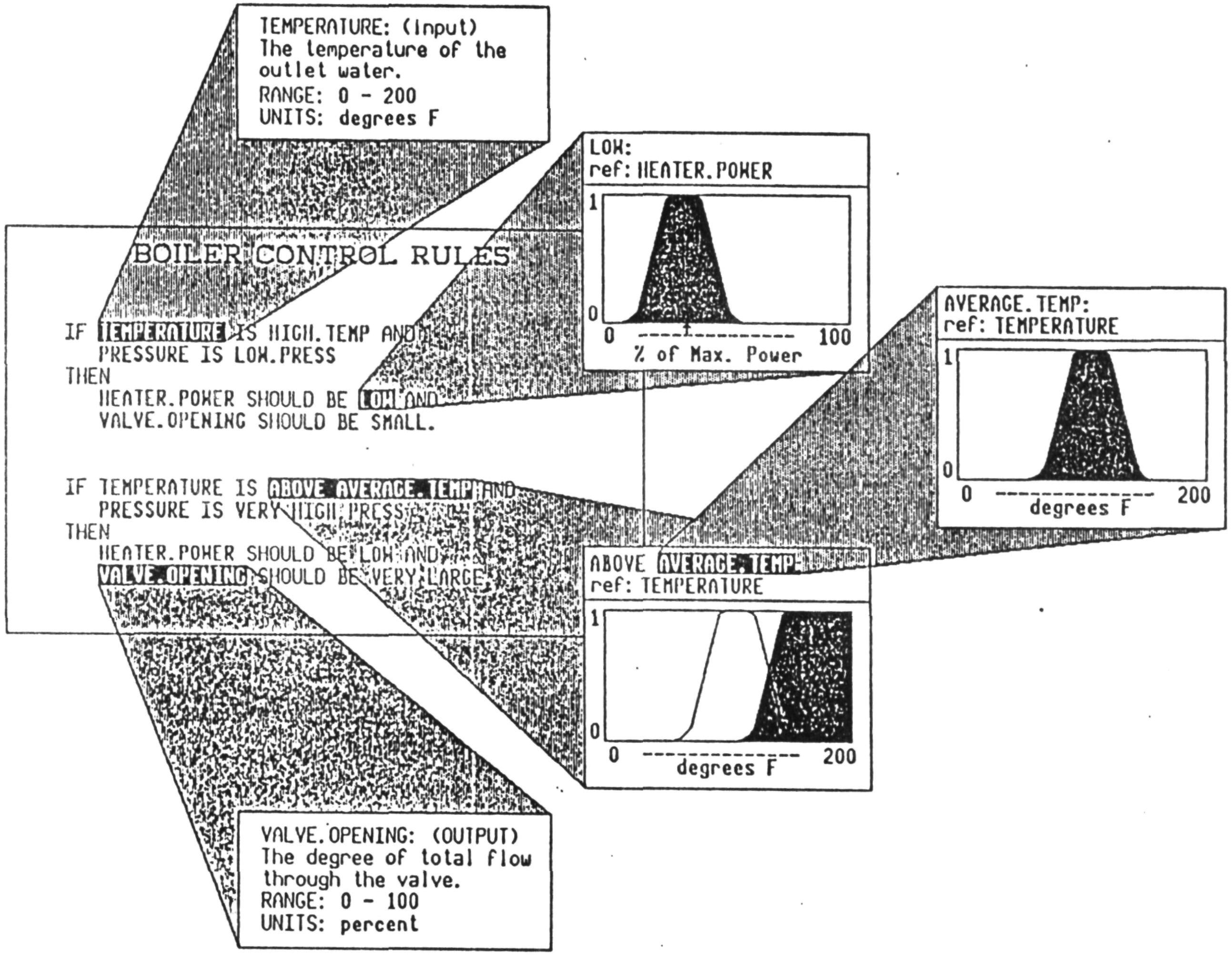
- Uniform graphical interface
- Screen-cut & text/graphics for documentation
- DOS executive provided

Graphical environment provides ease of knowledge acquisition

- Schematic representation of networks

Togai InfraLogic Inc.

ARTIFICIAL INTELLIGENCE ON A CHIP

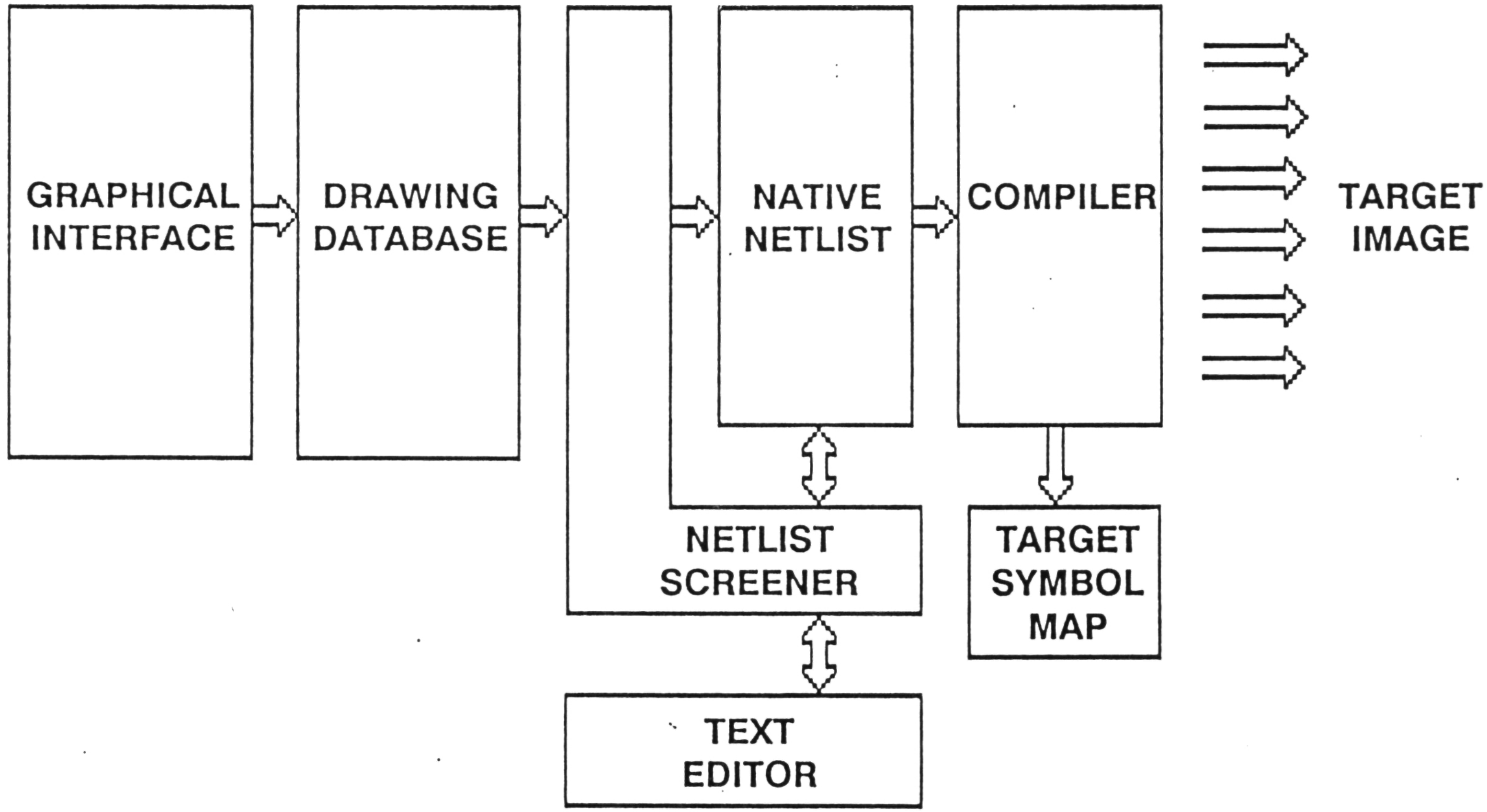


HYPERTEXT CONCEPT

VIEW12

ORIGINAL PAGE IS OF POOR QUALITY

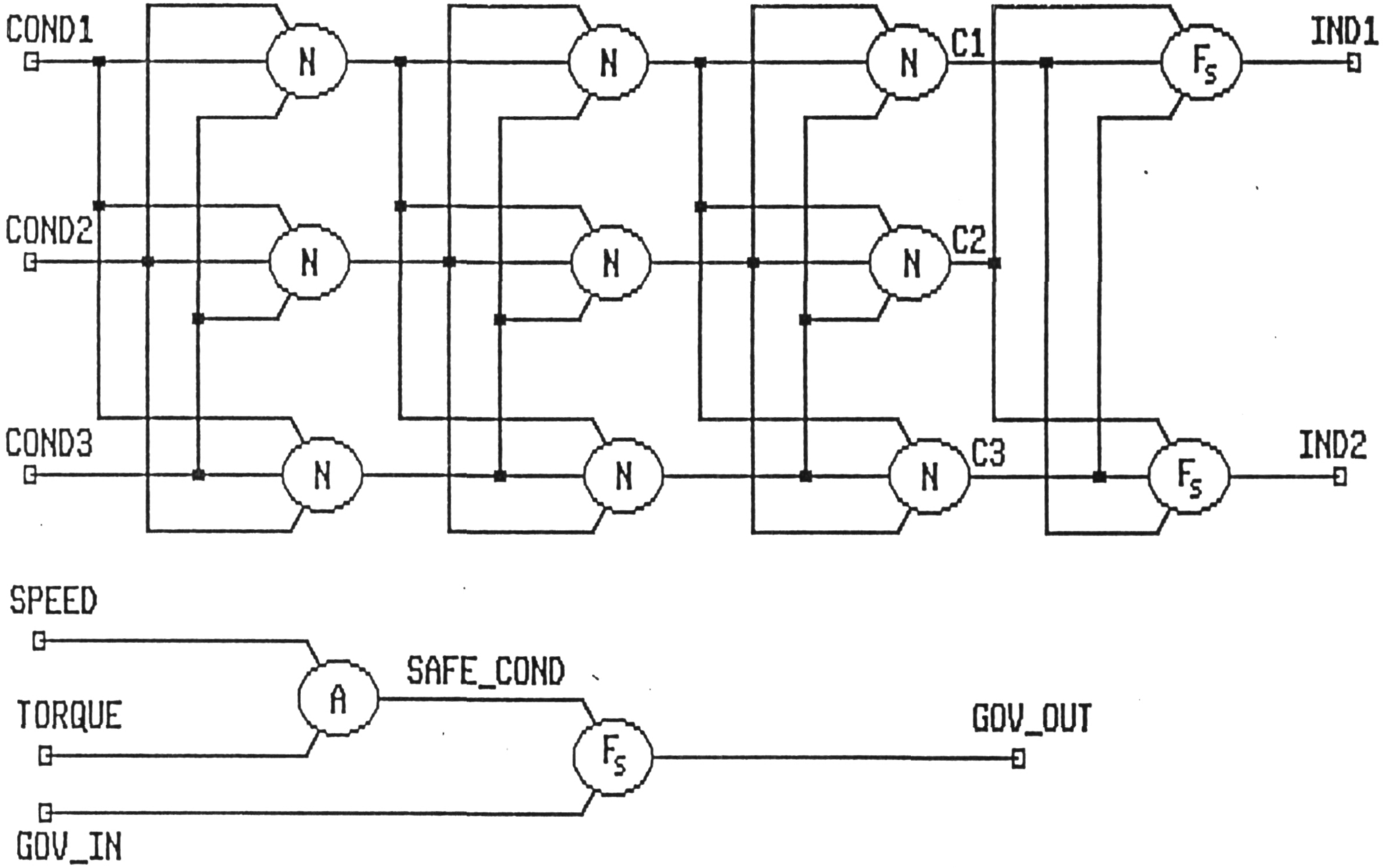
Togai InfraLogic Inc.
ARTIFICIAL INTELLIGENCE ON A CHIP



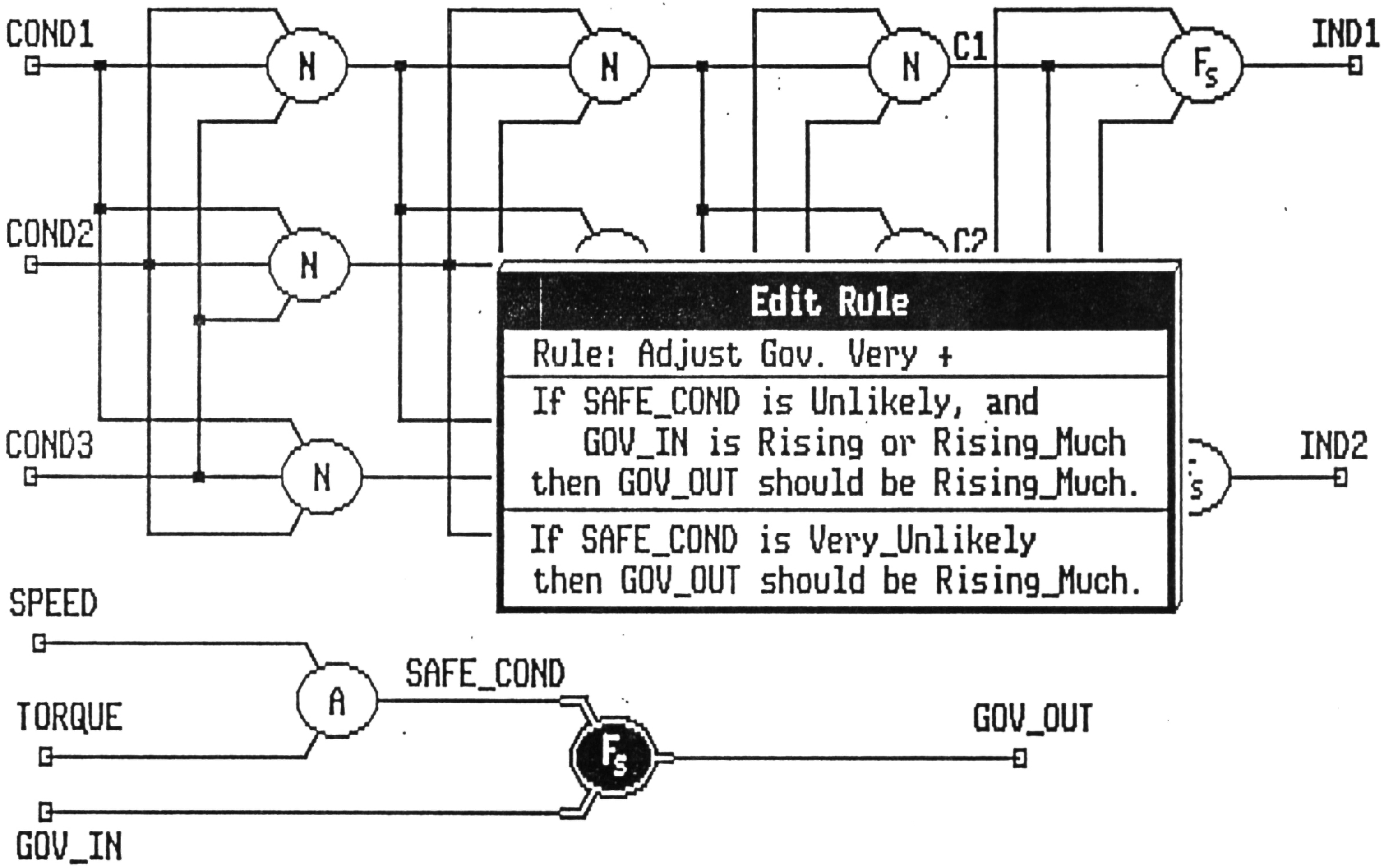
TIL SOFTWARE ENVIRONMENT

VIEW4

Togai InfraLogic Inc.
ARTIFICIAL INTELLIGENCE ON A CHIP

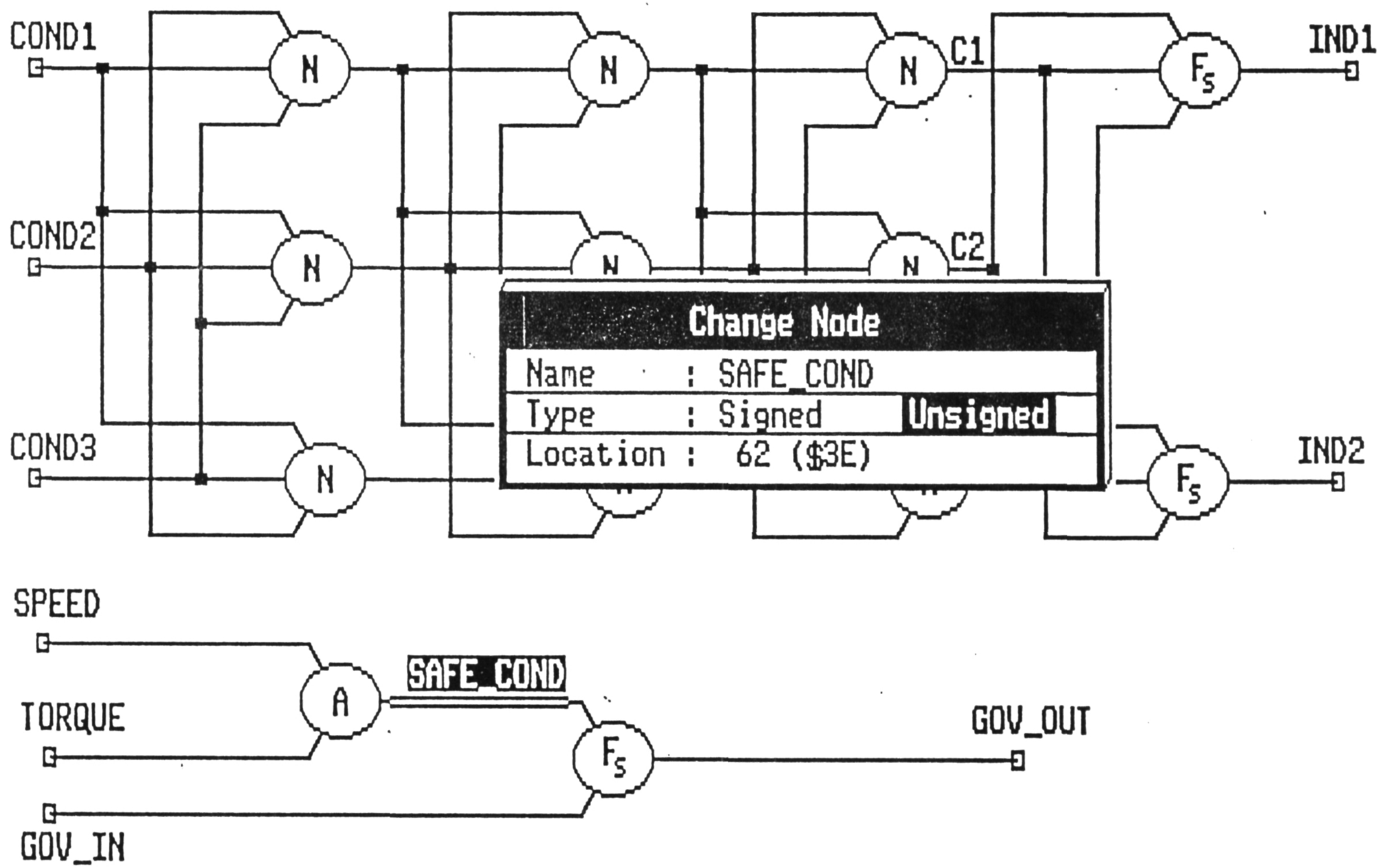


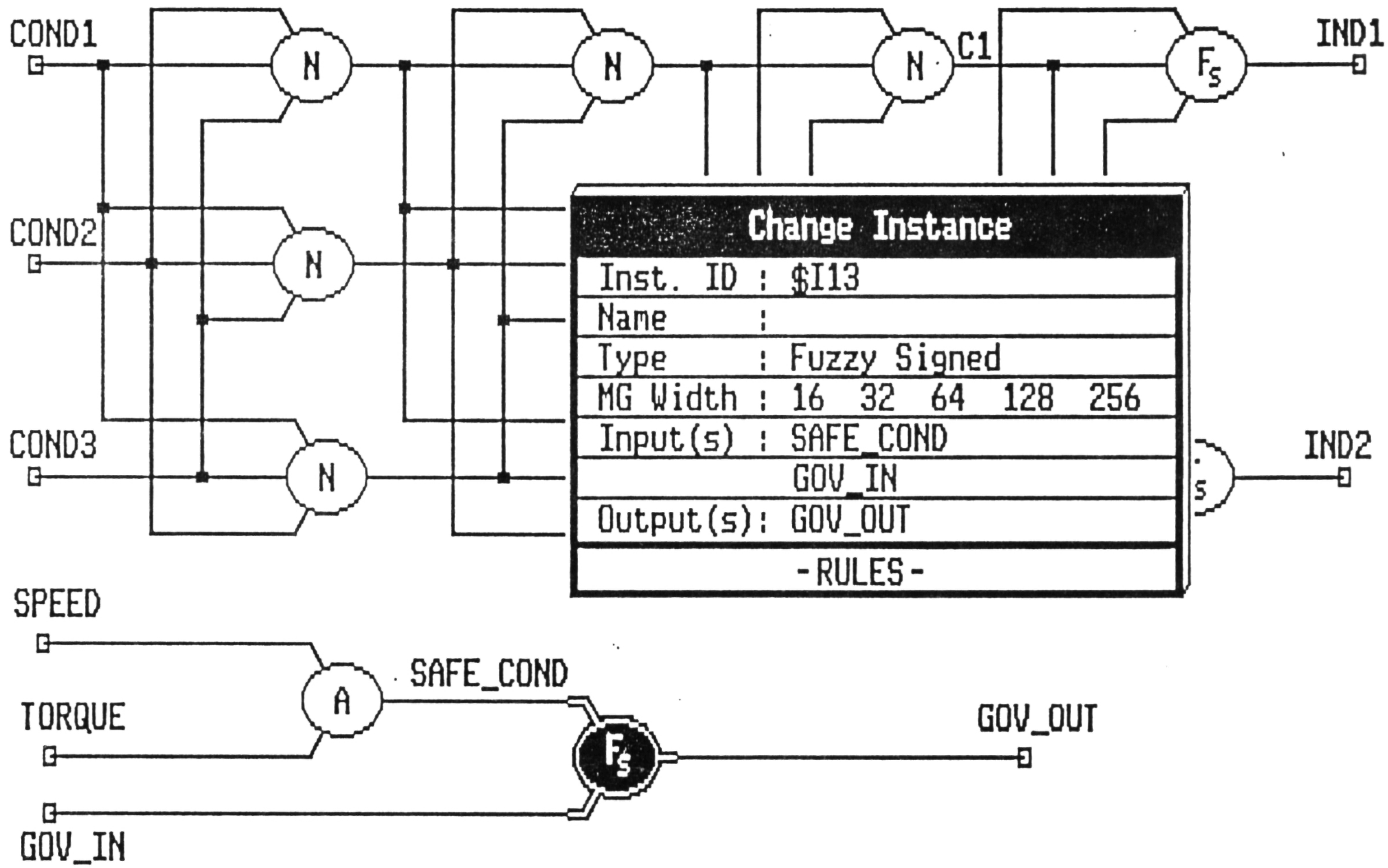
Togai InfraLogic Inc.
ARTIFICIAL INTELLIGENCE ON A CHIP



VIEW17

Togai InfraLogic Inc.
ARTIFICIAL INTELLIGENCE ON A CHIP




COND1
COND2
COND3
N
C1
F_S
IND1
IND2
Change Instance
Inst. ID : $I13
Name :
Type : Fuzzy Signed
MG Width : 16 32 64 128 256
Input(s) : SAFE_COND
GOV_IN
Output(s): GOV_OUT
-RULES-
SPEED
TORQUE
GOV_IN
A
SAFE_COND
GOV_OUT

VIEW16

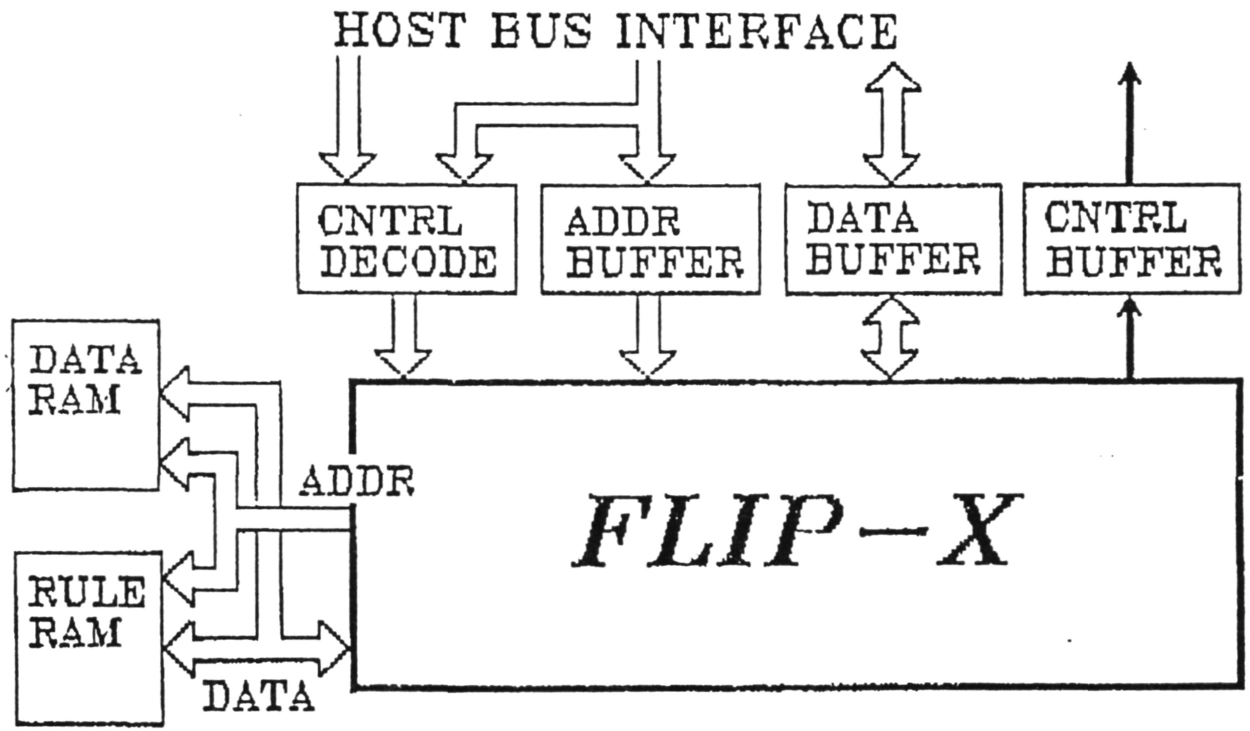
Togai InfraLogic Inc.
ARTIFICIAL INTELLIGENCE ON A CHIP

TIL HOST SYSTEM CONFIGURATION

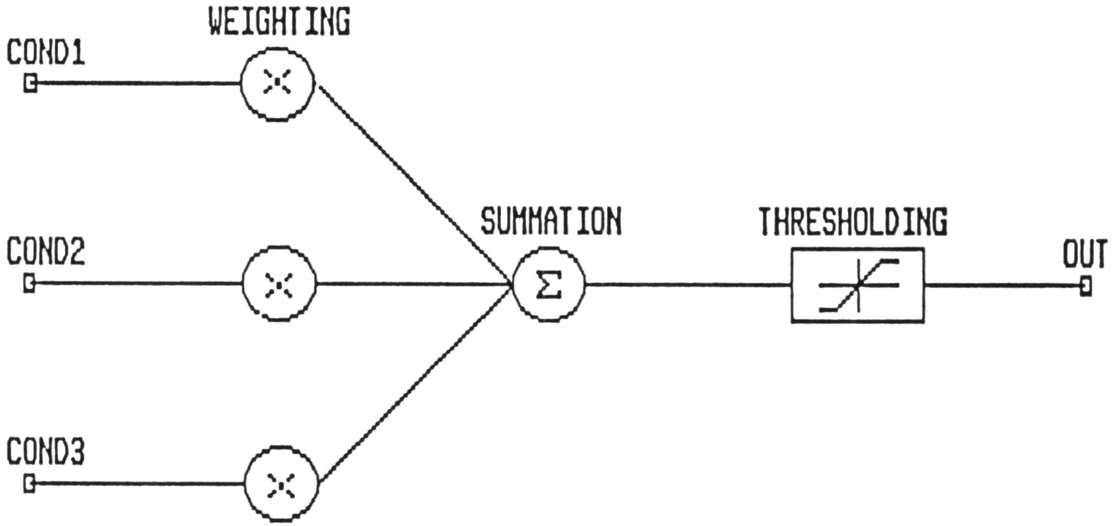
Host System : Togai InfraLogic 386 PC

Minimum Configuration:
- Processor: 16MHz 386
- Memory: 2 Mb
- Monitor: EGA color
- Disk Memory: 20Mb hard disk or greater
  - 1.2 Mb floppy disk drive
  - 360 Kb floppy disk drive
- Disk Operating System: DOS 3.2 or higher
- Slot Configuration: 6 AT slots

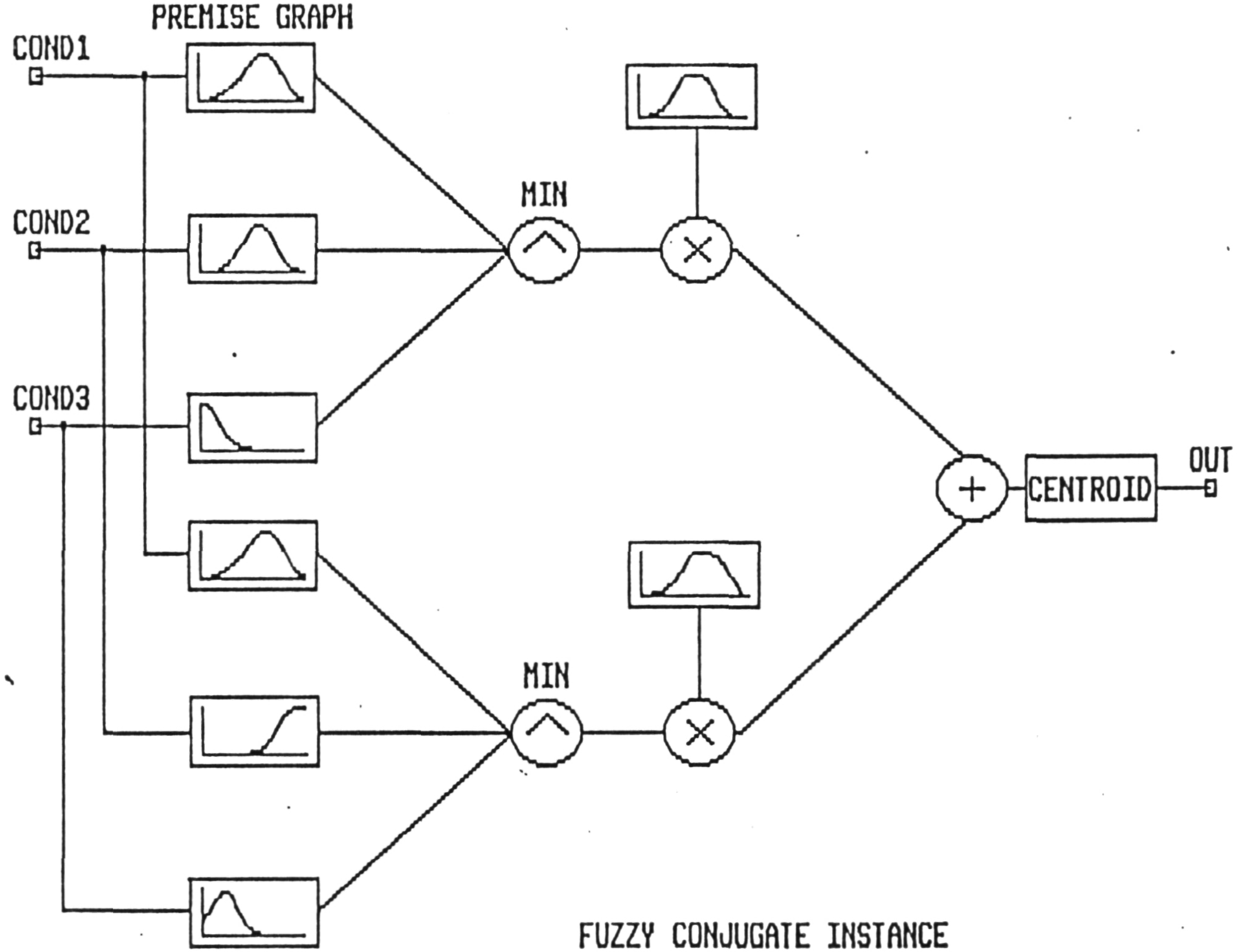
Additional Hardware;
- Togai InfraLogic Net-Processor Board
- Togai InfraLogic General Purpose I/O Board


HOST BUS INTERFACE
CNTRL DECODE
ADDR BUFFER
DATA BUFFER
CNTRL BUFFER
DATA RAM
RULE RAM
ADDR
DATA
FLIP-X

Togai InfraLogic Inc.
ARTIFICIAL INTELLIGENCE ON A CHIP



NEURAL COMPUTATIONAL NODE



FUZZY CONJUGATE INSTANCE

VIEW13

FUZZY NET

Real-Time Inferencing

Knowledge Acquisition Support Software

Flexibility of Connection & Membership Graphs

Eight Bit Computational & I/O Resolution

Trace Back & Storage

Reduction in Chip Level I/O

TIL NEURAL PARADIGM

Real-Time Processing

Flexibility of Connection

Eight Bit Computational & I/O Resolution

Weighting Accuracy 1%

All Nodes Resident & Visible

Reduction in Chip Level I/O

10^3 X Connectivity of Analog Solutions

Stable Across Voltage and Temperature

Togai InfraLogic Inc.
ARTIFICIAL INTELLIGENCE ON A CHIP

NOTABLE SYSTEM PERFORMANCE

CHIP

- Single Chip Net Processor
- Scalable SIMD Architecture
- Cascadable to MSIMD Architecture
- 10-20 MHz Clock Rate

FUZZY APPROXIMATE REASONING

- Processes up to 128 Production Rules Simultaneously
- Up to 256 Inputs and Outputs per Production Rule
- Greater than 20K FLOPs
- Greater than 200K Production Rule Evaluations per Second

NEURAL PROCESSING

- Processes up to 16 Neurons Simultaneously
- Greater than 65,000 Inputs per Neuron
- Transfer Function User Definable
- Greater than 2M Connections per Second

NOTES

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

S5-60

15306

N91-71355

p19

Hiroyuki Watanabe, Ph.D.

University of North Carolina
Chapel Hill, North Carolina

Dr. Watanabe received his M.S. in computer science from Indiana University in May 1978, and a second M.S. and Ph.D. from the University of Rochester in May 1980, and September 1983, respectively. In October 1983, he joined AT&T Bell Laboratories as a member of the technical staff and participated in two major research projects: the design of a VLSI fuzzy logic inference engine for real-time control, and development of an expert system for full custom VLSI design. Since August 1986, Dr. Watanabe has been a member of the Department of Computer Science at the University of North Carolina at Chapel Hill teaching VLSI design. As a consultant with the Microelectronics Center of North Carolina, he is continuing his research of fuzzy logic VLSI while designing a new chip with more than 600 000 transistors. He is also designing a software environment for programming and simulation of fuzzy rules, and is seeking a funding organization for this research. Dr. Watanabe's research interests include CAD for VLSI, VLSI design, fuzzy logic, applied AI and neural networks, with a particular interest in designing a VLSI architecture for AI application.

FUZZY LOGIC INFERENCE PROCESSOR - A CUSTOM VLSI DESIGN FOR SYSTEM INTEGRATION

Abstract

The VLSI implementation of a fuzzy logic inference mechanism allows the use of rule-based control and decisionmaking in demanding real-time applications such as robot control, and the area of command and control. The full custom CMOS VLSI is described. The chip is second generation of such design and has several design features which make its use realistic. These features include reconfigurable architecture, on-chip fuzzification and defuzzification, and memory and data-path redundancy for higher yield. The chip consists of 614 000 transistors, of which 460 000 are used for random access memory. For the fuzzy inference chip to be useful, we must package it into a system integrating hardware and software. We need to provide a user-friendly interface for control engineers. We are developing a system that combines graphic text inputs in a multiple-window environment. For rule set programming, a multiple-window environment provides editing and display facilities for the fuzzy rule sets, for fuzzy variables, and for the fuzzy set membership functions. Separate text and graphic windows interact with the user and display the developing system in various modes from different levels of abstraction. Simulation of the rule execution also can be displayed in graphic form.

Fuzzy Logic Inference Processor : A Full Custom Design for System Integration

Hiroyuki Watanabe
Wayne Dettloff†
James Symon
Phil Jacobsen
Russell Taylor
Ian Philp

Department of Computer Science
University of North Carolina
Chapel Hill, NC 27514
(919)962-1817

Microelectronics Center of North Carolina†
Research Triangle Park, NC 27709
(919)248-1874

March 26, 1988

abstract

The VLSI implementation of a fuzzy logic inference mechanism allows the use of rule-based control and decision making in demanding real-time applications such as robot control and in the area of command and control. The full custom CMOS VLSI is described. The chip is second generation of the design. It has several design features which make the use of this chip realistic. These features include reconfigurable architecture, on-chip fuzzification and defuzzification, and memory and data-path redundancy. The chip consists of 614,000 transistors of which 460,000 are used for RAM memory.

1 Introduction

Fuzzy logic based control uses a rule-based expert system paradigm in the area of real-time process control [4]. It has been used successfully in numerous areas including chemical process control, train control [12] cement kiln control [2], and control of small aircraft [5]. In order to use this paradigm of a fuzzy rule-based controller in demanding real-time applications, the

VLSI implementation of the inference mechanism has been an active research topic [9,10,11]. Potential applications of such a VLSI inference processor includes real-time decision-making in the area of command and control [3], control of the precision machinery [1], and robotic systems [6].

We have been designing a second-generation VLSI fuzzy logic inference engine on a chip. The new architecture of the inference processor has the following important improvement compared to previous work:

1. programmable rule set memory
2. on-chip fuzzifying operation – table lookup
3. on-chip defuzzifying operation – center of area algorithm
4. reconfigurable architecture
5. RAM redundancy for higher yield

The original prototype experimental chip (designed at AT&T Bell Labs) had minimal logic on chip. For example, it used ROM for the rule set memory which reduced its utility [10]. We are now designing a more realistic chip which has RAM for the rule set memory so that rules can be programmable. In addition to the fuzzy inference mechanism, the fuzzifying and defuzzifying operations are performed on chip. The new design has a reconfigurable architecture such that we can have either 51 rules, 4 inputs and 2 outputs, or 102 rules, 2 inputs and 1 output. These new design decisions render the new architecture realistic.

2 Fuzzy Set and Fuzzy Logic

Fuzzy set is based on a generalization of the concept of the ordinary set. In an ordinary set, we associate a characteristic function for each set. For example, we can define a set S with its characteristic function $f_s \rightarrow \{0,1\}$. Then, for all e in the universal set U,

$$\begin{array}{lll} e \in S & \text{if} & f_s(e) = 1, \\ e \notin S & \text{if} & f_s(e) = 0. \end{array}$$

Each element of the universe either belongs to or does not belong to the set S. In a fuzzy set, an element can be a member of the set with varying degree of membership. The associated characteristic function, therefore, returns any real number between 0 and 1, and it is termed as the membership function. For a fuzzy set F, we have an associated membership function $\mu_F(e) \rightarrow [0,1]$. For example, if element e is a member of fuzzy set F with degree 0.34, the associated membership function returns this value, $\mu_F(e) = 0.34$. If $\mu_F(e) = 0$, e is entirely outside of fuzzy set F, and if $\mu_F(e) = 1$, e is entirely inside of fuzzy set F. Fuzzy set is represented by a set of ordered pairs of an element u_i and its grade of membership:

$$F = \{(u_i, \mu_F(u_i))\}, \; u_i \in U$$

where U is a universe of discourse. Using a fuzzy set, we can represent and manipulate imprecise and vague concepts and data. For example, *approximately 100 km/h* is represented by the fuzzy

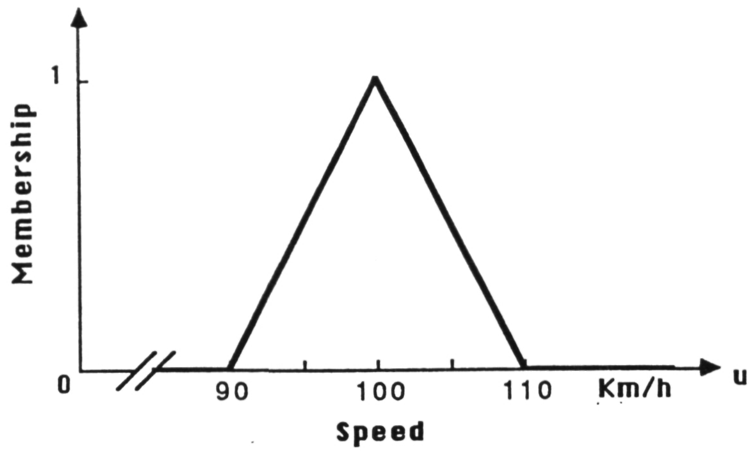


Figure 1: Approximately 100 km/h.

set whose membership function is shown in Figure 1. We can extend classical set theory by defining basic set theoretic operations over fuzzy sets. The following definition of intersection and union with fuzzy sets are suggested by Zadeh [13]. The set theoretic operations with fuzzy sets are defined via their membership functions. Let A and B be a fuzzy set, then union, intersection and complement of the fuzzy sets are defined as follows. The membership function of the intersection $C = A \cap B$ is defined by

$$\mu_C(e) = min(\mu_A(e), \mu_B(e)),\ e \in U.$$

The membership function of the union $D = A \cup B$ is defined by

$$\mu_D(e) = max(\mu_A(e), \mu_B(e)),\ e \in U.$$

The membership function of the complement $\neg A$ of A is defined by

$$\mu_{\neg A}(e) = 1 - \mu_A(e),\ e \in U.$$

In the traditional logic, one of the most important inference rules is *modus ponens*, that is

Premise	A is true
Implication	If A then B
Conclusion	B is true

Here, A and B are crisply defined propositions. We can construct a *fuzzy proposition* using a fuzzy set such as:

Current speed is *approximately 100 km/h.*

By introducing fuzzy propositions into modus ponens, we can generalize modus ponens. Let C, C', D, D' be fuzzy sets. Then the *generalized modus ponens* states:

Premise	x is C'
Implication	If x is C then y is D
Conclusion	y is D'

We can use different premises to arrive at different conclusions using the same implication. For example,

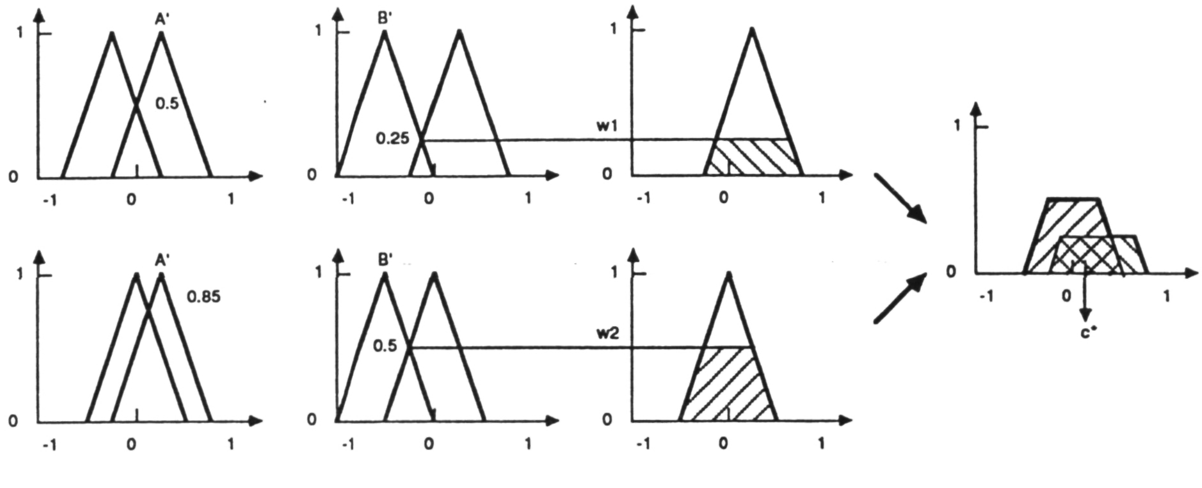


Figure 2: Inference.

Premise	Visibility is *slightly low*
Implication	If visibility is *low* then condition is *poor*
Conclusion	Condition is *slightly poor*

or

Premise	Visibility is *very low*
Implication	If visibility is *low* then condition is *poor*
Conclusion	Condition is *very poor*

The above inference is based on the compositional rule of inference for approximate reasoning proposed by Zadeh [14]. Suppose we have two rules with two fuzzy clauses in the IF-part and one clause in the THEN-part:

Rule 1: If (x is A_1) and (y is B_1) then (z is C_1),
Rule 2: If (x is A_2) and (y is B_2) then (z is C_2).

We can combine the inference of the multiple rules by assuming the rules are connected by OR connective, that is Rule 1 OR Rule 2 [10].

Given fuzzy proposition (x is A') and (y is B'), weights α_i^A and α_i^B of clauses of premises are calculated by :

$$\alpha_i^A = \max_X(\mu_{A' \cap A_i}(e)), \quad e \in X$$
$$\alpha_i^B = \max_Y(\mu_{B' \cap B_i}(e)), \quad e \in Y \quad for \quad i = 1, 2.$$

Then, weights w_1 and w_2 of the premises are calculated by :

$$w_1 = \min(\alpha_1^A, \alpha_1^B),$$
$$w_2 = \min(\alpha_2^A, \alpha_2^B),$$

Weight α_i^A represents the closeness of proposition (x is A_i) and proposition (x is A'). Weight w_i represents similar measure for the entire premise for the i^{th} rule. The conclusion of the first rule is

$$C_1' = \min(w_1, C_1),$$

The conclusion of the second rule is

$$C_2' = \min(w_2, C_2),$$

The overall conclusion C' is obtained by

$$C' = \max(C_1', C_2').$$

This inference process is shown in Figure 2. In this example, $\alpha_1^A = 0.5$ and $\alpha_1^B = 0.25$, therefore $w_1 = 0.25$. $\alpha_2^A = 0.85$ and $\alpha_2^B = 0.5$, therefore $w_2 = 0.5$.

3 Rule-based Controller

The usual approach for automatic process control is to establish a mathematical model of the process. However, this is not always feasible. In some cases, there is no proper mathematical model because the process is too complex or ill-understood. In other cases, experimenting with plants for construction of mathematical models is too expensive. In still other cases, the mathematical models are too complicated or computationally expensive and are not suitable for real time use. For such processes, however, skilled human controllers may be able to operate the plant satisfactorily. The operators are quite often able to express their operating practice in the form of rules which may be used in a rule-based controller. The rule based controllers model the behavior of the expert human operator instead of the process. The following is a rule from an aircraft flight controller [5]. This rule takes three inputs and has two outputs.

If (1) The rate of descent is *Positively Medium*,
(2) The airspeed is *Negatively Big* (compared to the desired airspeed),
(3) The glide slope is *Positively Big* (compared to the desired slope).
Then (1) change engine speed by *Positively Big*, and
(2) change elevator angle by *Insignificant Change*.

The expressions, *Positively Medium*, *Positively Big*, *Insignificant Change*, and others represent imprecise amounts. They represent intuitive feel of the expert human controller. They correspond to the imprecise expressions used by the expert for communicating a rule of thumb. They are represented by using fuzzy sets and their associated membership functions.

The fuzzy set, such as *Positively Medium* is represented by the membership function over an appropriate universe of discourse such as revolutions per minute (rpm). The possible definitions of fuzzy sets are shown in Figure 3. The control rules are encoded using typically 10 to 70 rules. The Control is performed based on the fuzzy inference mechanism described in Section 2 and Figure 2. In controlling a process, all of the rules are compared to the current inputs (observations) and fired. The actions (THEN-part) of each rules are weighted by how close its IF-part matches the current observation. In the example of Figure 2, a rule has two inputs and

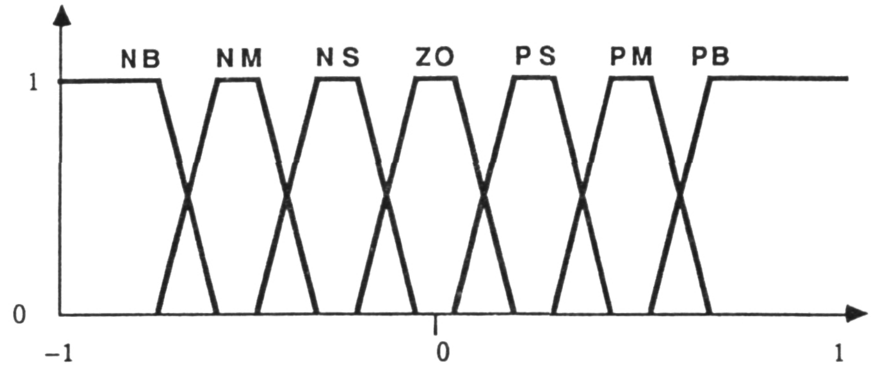


Figure 3: Typical fuzzy sets.

a single output. The weights are represented by w_1 and w_2. The results of firing of each rule are then combined by superimposing them. The final result which is supplied to a controller should be a crisp number rather than a fuzzy set, therefore we need to perform a defuzzifying operation. This is computed by taking a center of area under the fuzzy membership function of the final result. Even though each individual rule is an incomplete rule of thumb, the results of firing each rule are properly weighted and combined and the final result represents reasonable compromise.

In order for VLSI implementation of fuzzy inference to be useful, a fair amount of pre-processing (fuzzifying) and post-processing (defuzzifying) must be performed on chip. The AT&T prototype chip assumed that both of these processes are performed by the host-processor. However, the inference processing is too fast for fuzzifying and defuzzifying to take place off-chip by a host processor. This assumption burdened the host processor and nullified the advantage of VLSI implementation of the inference mechanism.

4 Chip Architecture and Implementation

The process controller system is configured as in Figure 4. The VLSI implementation is done with four components; a fuzzyer, a rule memory, an inference mechanism, and a defuzzifier on a single chip. Each input and output data item is 6 bits. This fits well with available A/D and D/A converters. In addition, our chip will communicate with a host processor. The chip has three stage pipelining architecture. The pipeline consists of IF-part, THEN-part, and defuzzifier.

We considered the size of the fuzzy set and the grade of fuzziness for practical use. In most cases, a fuzzy variable has three to sixteen elements and the grade of fuzziness has three to twelve levels [5,8]. In this chip implementation, the universe of discourse of a fuzzy set is a finite set with 64 elements (i.e. 6 bits). The membership function has 16 levels (i.e. 4 bits). That is, 0 represents no membership, 15 represent full membership, and other numbers represent points in the unit interval [0, 1]. A fuzzy membership function is, therefore, discretized using 64 numbers of 4 bit; that is 256 bits of memory storage. The representation of a fuzzy set is as follows:

u_0	u_1		u_i		u_{63}
0000	0011	...	$\mu_F(u_i)$	...	0000

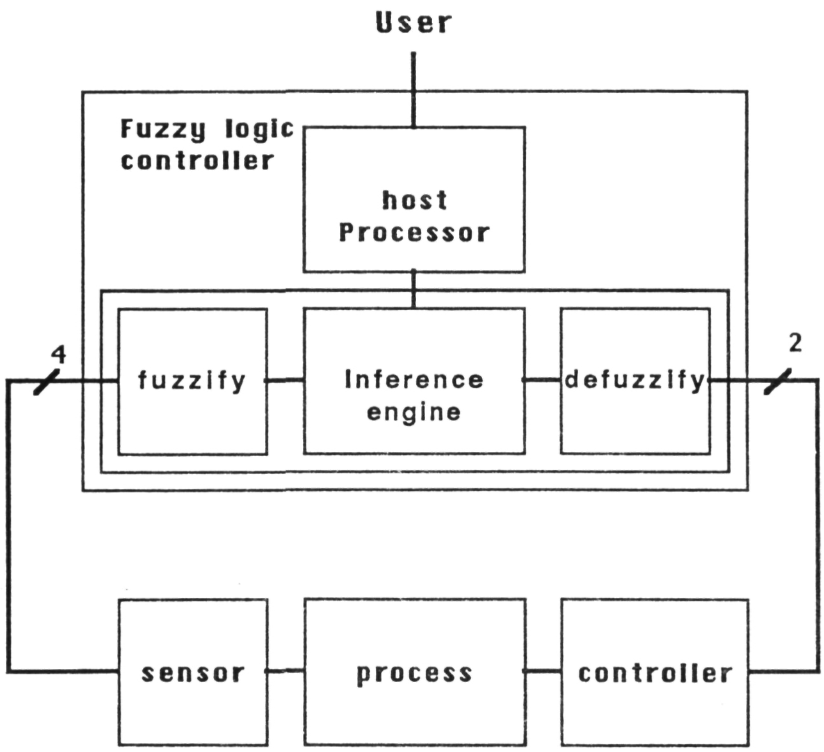


Figure 4: Fuzzy logic controller.

Fuzzifying is done using a table look-up. For each observation (i.e. input stream), we store a table of the membership function normalized at the center of the horizontal axis. That is, the full membership is at the center. According to an input value, the membership function is shifted. The chip can produce 64 different membership functions from a single stored pattern. The membership function can be associated with a predicted measurement error of a sensor. If we do not need fuzziness in the observed value, we can store a pulse function, that is only one entry has membership 1 and all the other entries have 0's. The result of the fuzzifying is broadcasted to all of the rules. In the actual chip implementation, the content of the table is not shifted. Rather a starting address for table look-up is shifted according to an observation input.

The chip is re-configurable. A control system can take four inputs and produce two outputs or take two inputs and produce one output according to an application. With the first configuration, we can have 51 rules on a single chip. Each rule has four clauses in the IF-part and two actions in the THEN-part.

If A and B and C and D
Then Do E, and
Do F.

With the second configuration, we can execute 102 rules using a same data-path. Each rule has two clauses in the IF-part and one action in the THEN-part.

If A and B Then Do E,
If C and D Then Do F.

A data-path is assigned for each rule, therefore all of 51 or 102 rules are executed in parallel. There are only two basic units; they are a parallel minimum unit and a parallel serial unit. The former performs the intersection operation on fuzzy sets, and the latter performs the union operation. The configuration of the If-part of the data-path is shown in figure 5. The data-path can execute one rule with 4 if-clauses or two rules with 2 if-clauses. Four pairs of min/max units compute the weight α's for each clause. The min elements organized as a binary tree compute weights w of the premise which is the minimum of all α's. In the 51 rule configuration, the last two minimum units compute the same weight w_i. In the 102 rule configuration, streams of 1's are supplied and these two min elements behave as delay elements. The control of configuration is done by setting a bit in the status register from the host computer. Defuzzifying is done by computing a center of area (COA) under the final membership function. Denoting the final fuzzy subset as A, the COA algorithm computes the following:

$$c^* = \frac{\sum_{n=0}^{63} n \cdot \mu_A(n)}{\sum_{n=0}^{63} \mu_A(n)}$$

Since each element of the universe is processed serially, we can substitute multiple addition for multiplication in the above computation. The data sequence from the THEN-part is produced starting from the most significant data point as follows:

$$\mu_A(63),\ \mu_A(62),\ \ldots,\ \mu_A(1),\ \mu_A(0).$$

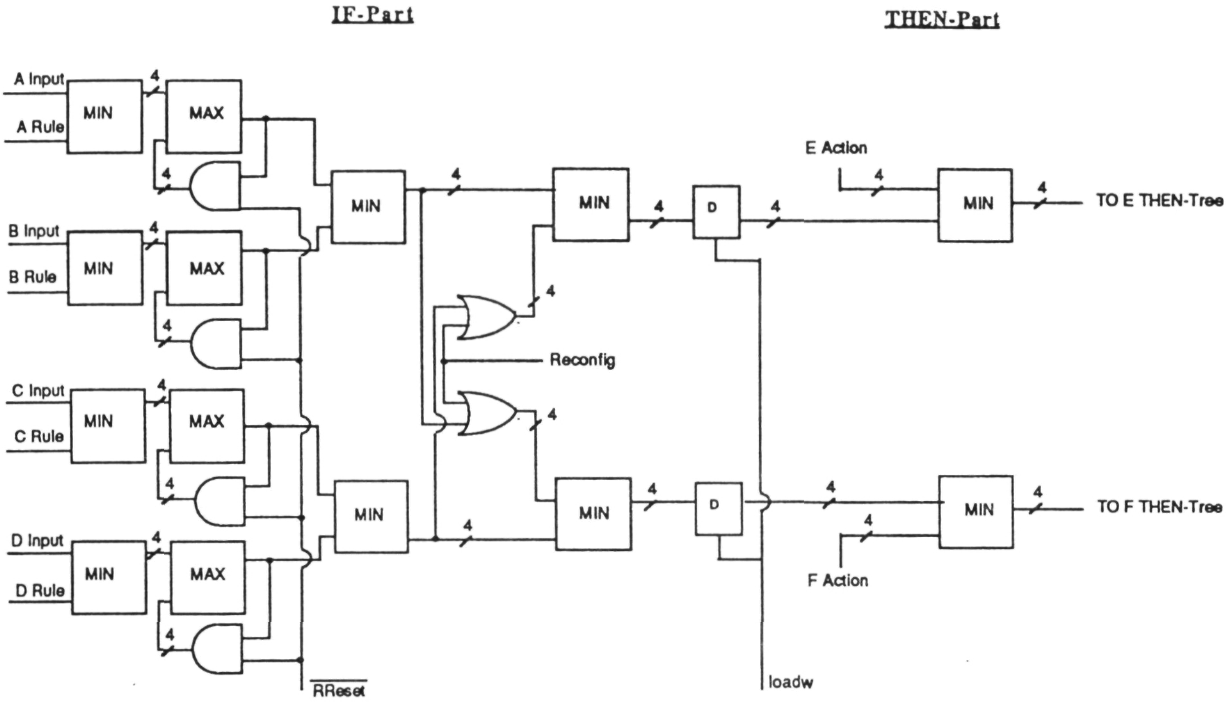


Figure 5: Reconfigurable data-path for rule execution.

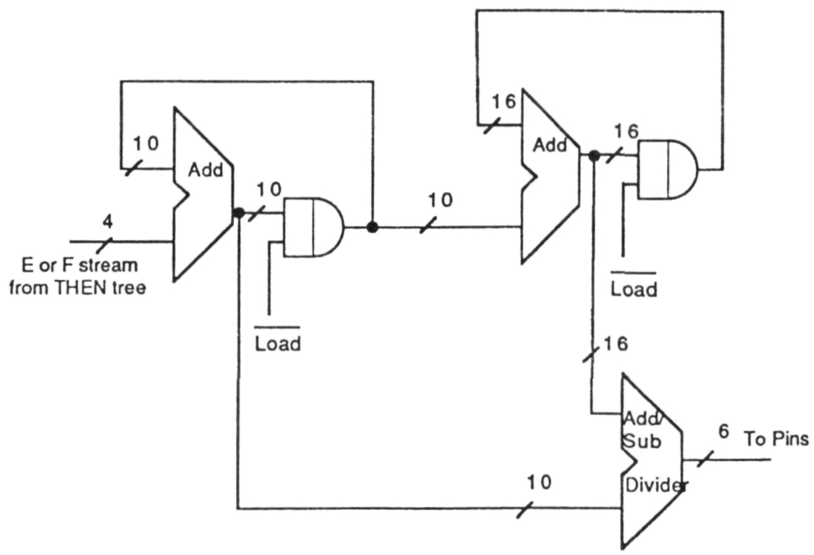


Figure 6: Defuzzifier circuit.

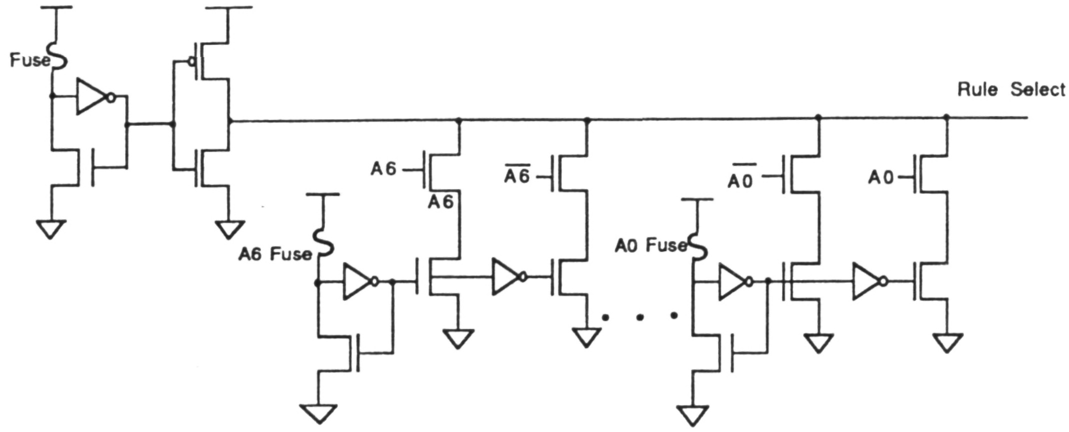


Figure 7: Redundancy

Two adders and two registers are used as shown in Figure 6. The numerator is computed by the first adder and denominator is produced by the second adder. The denominator is computed as by repeated addition of the result of the first adder by the second adder which computes the following formula.

$$\begin{aligned}\sum_{n=0}^{63} n\mu_A(n) &= \mu_A(63) + \\ & \quad \mu_A(63) + \mu_A(62) + \\ & \quad \mu_A(63) + \mu_A(62) + \mu_A(61) + \\ & \quad \vdots \\ & \quad \mu_A(63) + \mu_A(62) + \mu_A(61) + \cdots + \mu_A(0).\end{aligned}$$

In order to achieve higher yield, we allocated 51 data-paths on the chip, and non-functioning memory units and data-paths can be isolated from the rest of the chip. The isolation is achieved by blowing a fuse using laser technology. Each pair of a memory unit and a data-path can be reprogrammed to any other address also by blowing a fuse. This allows a continuous addressing of memory/data-paths after removal of a defective unit from a chip. The schematic diagram for address removal and re-programming circuit is shown in Figure 7.

The host processor down loads the rule set and table for fuzzification at start up time. The fuzzy processor looks like a static RAM chip to the host processor. The RAM system, however, only has a row decoder and does not have a column decoder. A user can address each row (corresponds a clause/action of a rule) by a memory address register. Each column is addressed by a shift register because data are accessed sequentially. The last address is reserved and mapped to the status register. This register controls the configuration of data-paths and operational modes (load, run, or test). Fuzzification tables have their own memory address and loaded similarly as rule memory.

The chip is designed for a 1 μm N-well CMOS process of MCNC [7]. It uses non-overlapping

Die Size	$7750\mu \times 9080\mu$
No. Transistors	614K (470K RAM)
No. Pins	84 (16 Power/GND)
Package Type	PGA (Standard Pad Frame)
Clock Frequency	40 MHz @ 70°C
Power Supply	3.0 –3.3 v
Power (Est.)	600mW
Interface	TTL Compatible
Modes	4 inputs/2 outputs/51 rules, 2 inputs/1 outputs/102 rules, test
Redundancy	Laser Programmable
Process	1 μm N-well CMOS
Gate Length/t_{ox}	1.0μm/22.5nm
Poly/Metal 1/Metal 2	2.6/2.6/4.0μm

Table 1: Summary of circuits

two phase clocking scheme. The chip is designed with a target operational speed of 40MHz. The chip consists from approximately 614,000 transistors of which about 470,000 are used to form the static RAM system. The die size is 7750μm by 9080μm, and is packaged in a standard pin grid array with 84 pins. The supply voltage is 3.0–3.3 v. Table 1 summarizes the process, device specifications and primary architectural features. Figure 8 shows the layout map of the chip.

5 System Integration

For the fuzzy inference chip to be useful we must package it into a system integrating hardware and software, hence development of hardware and software must be coordinated. We need to provide a user friendly interface to control engineers. We have performed a substantial work in development of software system. Hardware side of the system integration is in a preliminary design stage.

5.1 Hardware System

For the hardware side, we will package the VLSI chip into a **single board system**. The single board system should be bus compatible with widely available personal computers or workstations. Potential candidates are: 1) IBM PC/AT, 2) Sun workstation, 3) IBM Personal System II, 4) Apple MachIntosh II. At this moment, we believe either IBM PC/AT or Sun workstation is most suitable for our purpose. IBM PC/AT is widely available and is used in factory automation. On the other hand, we have extensive software on Sun worksation.

Layout Map

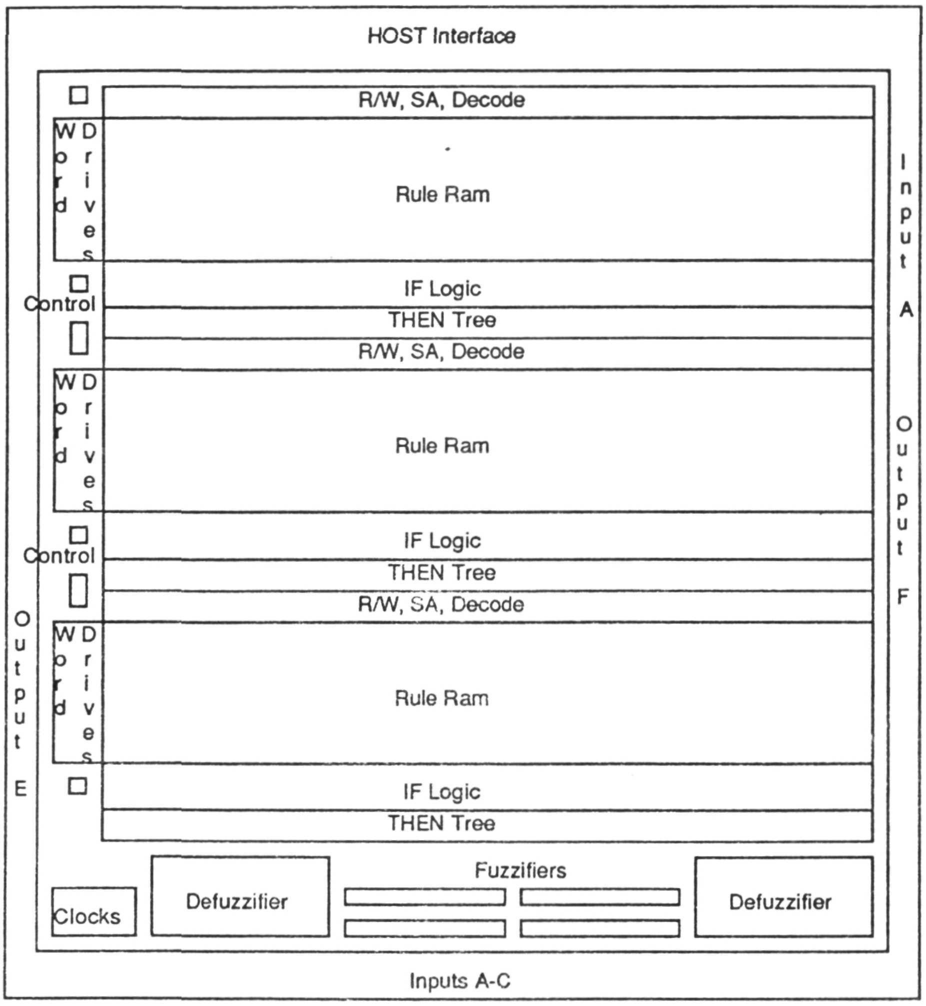


Figure 8: Layout map

The single board system consists of a VLSI fuzzy logic inference processor chip, logic for a standard bus interface, A/D converters for inputs, D/A converters for output, and glue logic. For applications requiring more rules, we can combine multiple fuzzy chips into one inference processing system. We would only need a small amount of extra glue logic and chip control. Overall the single board system is fairly modest and should be easy to construct.

5.2 Software system

For software system integration, we need a **programming environment** for developing the control rules, and software to communicate and drive the fuzzy logic inference board from a host processor. We have been developing a system that combines graphic input and text input in a windowed environment using X window system. Window environment is useful for editing of rule set, and graphic representation of simulation of rule set execution.

5.3 Programming Environment

As discussed above, the chip's output is driven by a set of IF-THEN rules. A rule set should be easy to develop, test and load into the chip. Our programming environment allows a user easily to describe a rule set which represents operating practice in the system that the chip will control. The user must be able to define membership functions and assign them to IF and THEN clauses of the rules. Fuzzy variables which will take on input values during chip operation must also be assigned membership functions for the fuzzifying process. The environment allows easy simulation and testing of the rule set. The simulated execution is displayed graphically for ease of debugging and refinement of the rule set. Finally, the rule set will be down-loaded to the Fuzzy Logic Board.

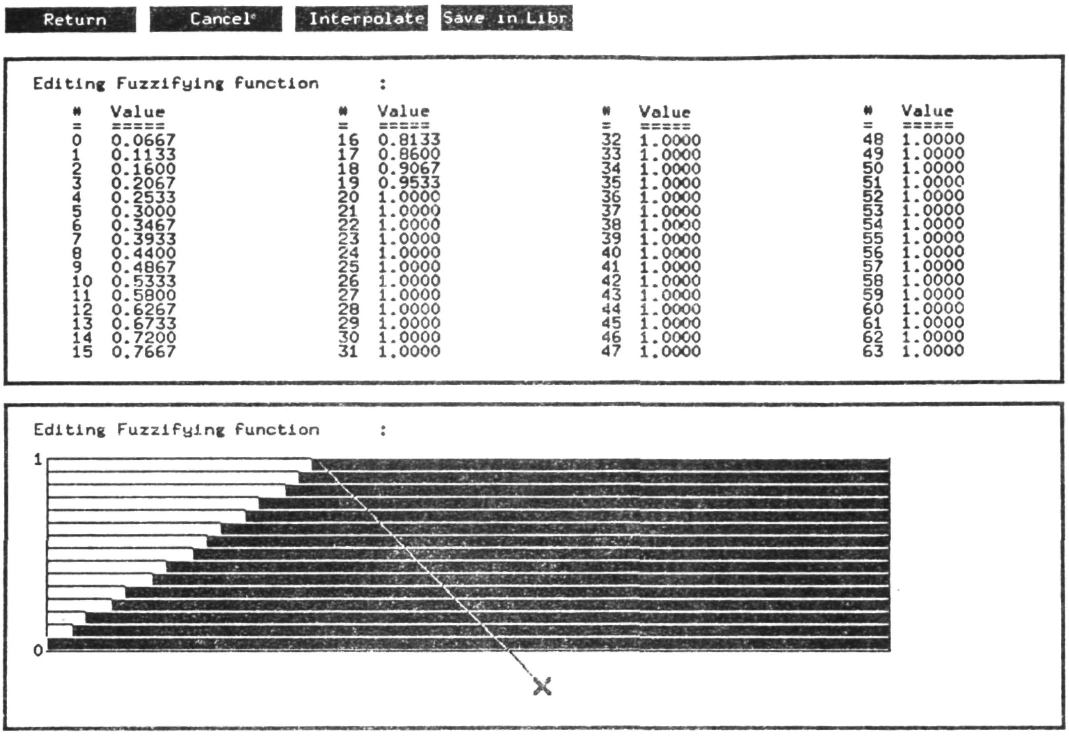
5.3.1 Editors

For rule set programming, a multiple window environment provides editing and display facilities for the fuzzy rule sets, for fuzzy variables, and for the fuzzy set membership functions used in both the fuzzifying process and the representation of the rule clauses. Separate text and graphic windows interact with the user and display the developing system in various modes and from different levels of abstraction.

Working in the editors, the user may proceed sequentially or select randomly among the items to be defined. Automatic sequential entry allows fast initial setup of prototype rule systems. Correction and modification require random access.

For each of the editors, (fuzzy set membership functions, fuzzy variables, and the fuzzy rule set), a text window and a graphics window are available and may be displayed simultaneously. Editing may proceed by text input to the text window, or by mouse and keyboard input to the graphics window. As changes are made in one window, the corresponding changes will appear in the other window as appropriate to that mode.

A fuzzy set membership function is represented internally as 64 discrete numbers, each specifying the membership at one point in the universe of discourse. Graphic input of the corresponding shape may be made by line segments which are immediately translated into the step function of discrete values. Figure 9 shows an actual screen of Sun workstation perfoming



#	Value	#	Value	#	Value	#	Value
0	0.0667	16	0.8133	32	1.0000	48	1.0000
1	0.1133	17	0.8600	33	1.0000	49	1.0000
2	0.1600	18	0.9067	34	1.0000	50	1.0000
3	0.2067	19	0.9533	35	1.0000	51	1.0000
4	0.2533	20	1.0000	36	1.0000	52	1.0000
5	0.3000	21	1.0000	37	1.0000	53	1.0000
6	0.3467	22	1.0000	38	1.0000	54	1.0000
7	0.3933	23	1.0000	39	1.0000	55	1.0000
8	0.4400	24	1.0000	40	1.0000	56	1.0000
9	0.4867	25	1.0000	41	1.0000	57	1.0000
10	0.5333	26	1.0000	42	1.0000	58	1.0000
11	0.5800	27	1.0000	43	1.0000	59	1.0000
12	0.6267	28	1.0000	44	1.0000	60	1.0000
13	0.6733	29	1.0000	45	1.0000	61	1.0000
14	0.7200	30	1.0000	46	1.0000	62	1.0000
15	0.7667	31	1.0000	47	1.0000	63	1.0000

Figure 9: Graphic editor

this task. The function is given a name such as *positive medium*. Alternatively, a membership function may be a prededined shape such as a triangular.

A library of fuzzy set membership functions gives the programmer the option of using predefined terms for the rule set clauses and fuzzifying functions. Without the need for extensive initial definition of terms, prototyping can progress quickly to the simulation stage. The system may then be fine-tuned through custom redefinition of terms. Predefined fuzzy set membership functions may also be associated with application derived terminology without the need for customized function shape specification. Additions and deletions may be made in the library.

A fuzzy variable is the internal representation of some input or output such as airspeed, glide slope, or elevator angle. For processing by the fuzzy system, a single value is represented by a membership function over a universe of discourse. Thus a fuzzy variable must be associated with a membership function which will fuzzify an input value or represent the output of a rule for subsequent output value determination. Using the editor, the associated function may be layed out in the graphics window or an existing membership function name may be specified in the text window. The corresponding graphic shape will then appear in the graphics window.

The rule editor has a structured text editor. The user fills in the blanks and is prompted

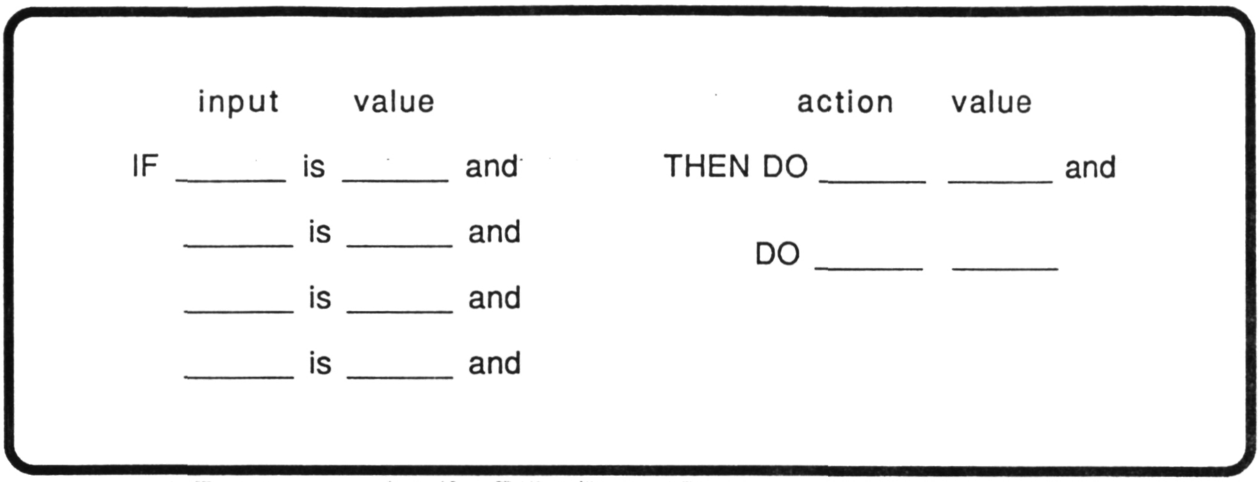


Figure 10: Rule editor – text window.

at the next blank. See Figure 10. The user may move the cursor to any blank and may select at random the rule currently being edited. As blanks are filled in, the corresponding graphic shapes will appear in the graphics window. The window configurations on Sun workstation are shown in Figure 11.

5.3.2 Simulation

The development of control rules is experimental in nature; a trial and error approach is customary. Simulation of the rule set system is therefore required. This includes offline, software simulation of the behavior of the chip as well as interaction with a program simulating the process to be controlled. These simulation processes are integrated with the rule editing facilities. The rule set programmer makes changes and views their effects without delay or exiting from the system.

The system graphically displays the inference process of the simulated chip execution within the system windows. This facilitates debugging and refinement of the rule set. Rule by rule analysis of the simulation is possible as well as monitoring overall behavior. The user selects a subset of the rules. This subset, which may be one, some, or all of the rules, can be fired one at a time or simultaneously. The effect on the chip output is displayed in a separate window. Any subset may also be 'unfired', or deleted after firing of some, or all of the rules. The system then displays the intermediate or final output that would result absent that rule or subset of rules. Again, this unfiring may be done stepwise or simultaneously.

The system makes the output available at interprocess communication sockets, and similarly will accept input variable values at sockets. A simulation of a process to be controlled by the chip may thus be controlled directly from the rule set programming environment. The actual operation of the current rule set on the controlled process may be monitored and the rule set

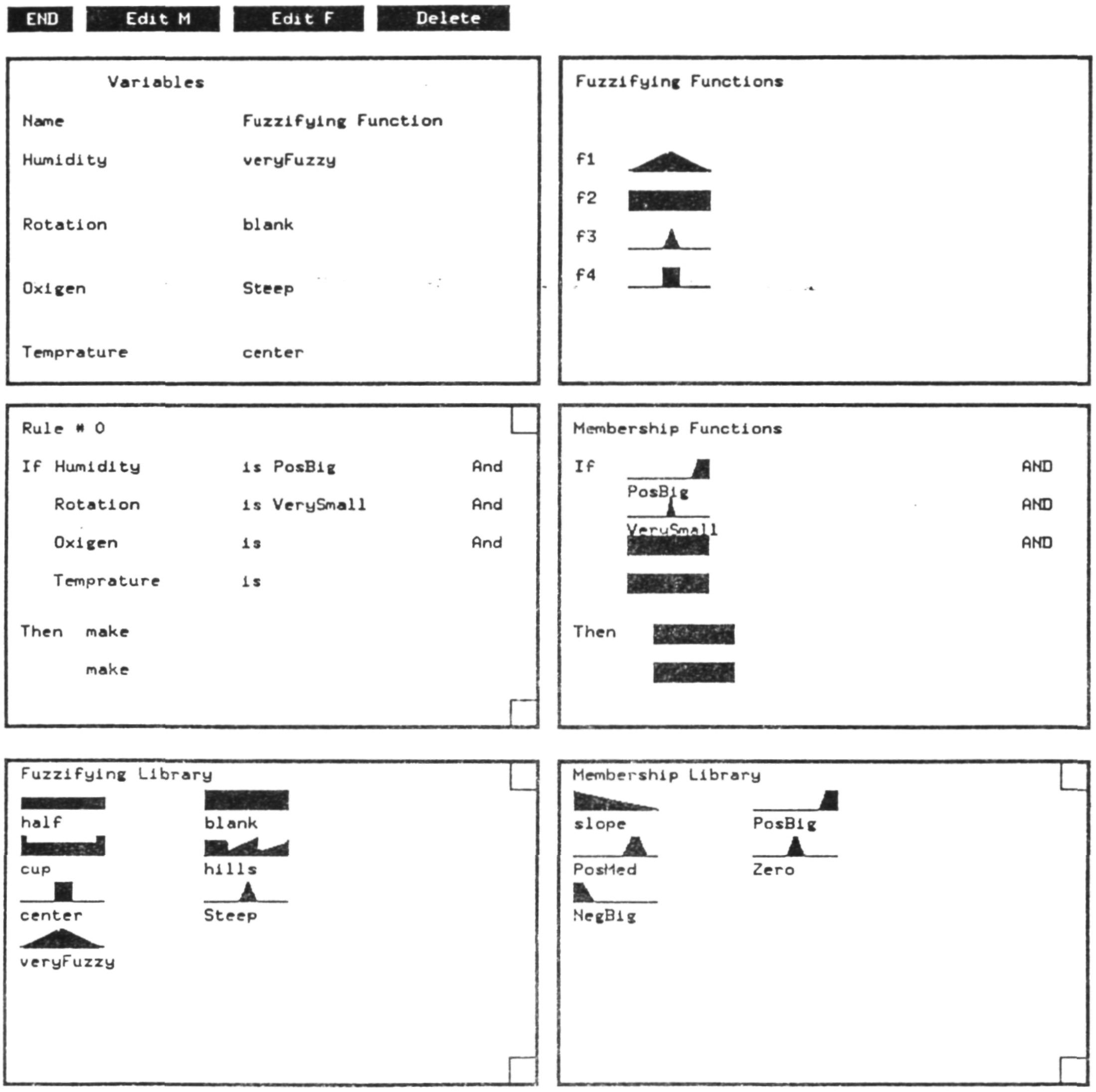


Figure 11: Window configuration.

changed immediately in response to this simulation.

5.4 Device Driver

Driving the chip is fairly simple. It is done by down loading a rule set and setting the chip to run mode. At execution time, the chip can communicate with A/D and D/A converters either directly or through a host. To the host, the fuzzy logic chip looks like a static RAM chip. It has the usual R/W and enable pins. Down-loading of the rule is done using address and data registers.

6 Acknowledgement

The defuzzifying circuits were designed by Jeff Hultquist and Jih-Fang Wang of the University of North Carolina at Chapel Hill. Kathy E.Yount of MCNC assisted for designing VLSI layout and producing figures for publication. Research reported here is supported in part by Microelectronic Center of North Carolina through MCNC Design Initiative Program, by the Office of Naval Research (Contract No. N00014-86-0680) and University of North Carolina Junior Faculty Development Fund.

References

[1] Burg, B., L. Foulloy, J. C. Heudin, and B. Zavidovique, "Behavior Rule Systems for Distributed Process Control," *Proc. of 2ed Conf. on AI Applications*, pp. 189-203, December 1985.

[2] Holmblad, L. P. and J. J. Ostergaard, "Control of a Cement Kiln by Fuzzy Logic," in *Fuzzy Information and Decision Processes*, M. M. Gupta and E. Sanchez (Eds), pp. 389-399, 1982.

[3] Kawano, K., M. Kosaka, and S. Miyamoto, "An Algorithm Selection Method Using Fuzzy Decision-Making Approach," *Trans. Society of Instrument and Control Engineers*, Vol. 20, No. 12, pp. 42-49, 1984. (in Japanese)

[4] King, P. J. and E. H. Mandani, "The Application of Fuzzy Control Systems to Industrial Processes," *Automatica*, Vol. 13, No. 3, pp. 235-242, 1977.

[5] Larkin, L. I., "A Fuzzy Logic Controller For Aircraft Flight Control," in *Industrial Applications Of Fuzzy Control*, M. Sugeno (Ed), pp. 87-103, 1985.

[6] Murakami, S., F. Takemoto, H. Fujimura, and E. Ide, "Weld-line Tracking Control of Arc Welding Robot Using Fuzzy Logic Controller," *Proc. of 2nd Inter. Fuzzy Systems Association Congress*, pp. 353–357, July 1987.

[7] Sharma, D., S. Goodwin-Johansson, D. S. Wen, C. K. Kim, and C. M. Osburn, "A 1μm CMOS Technology with Low Temperature Processing," *Extended Abstracts of 171 meetings of the Electrochemical Society*, Vol. 87-1, pp. 213-214, May 1987.

[8] Sugeno, M. and Murakami, K., "Fuzzy Parking Control of Model Car," *Proc. the 23rd IEEE Conf. Decision and Control*, December 1984.

[9] Togai, M, and S. Chiu, "A Fuzzy Logic Accelerator and a Programming Environment for Real-Time Fuzzy Control," *Proc. of 2nd Inter. Fuzzy Systems Association Congress*, pp. 147-151, July 1987.

[10] Togai, M. and H. Watanabe, "An Inference Engine for Real-time Approximate Reasoning: Toward an Expert on a Chip," *IEEE EXPERT*, Vol. 1, No. 3, pp. 55-62, August 1986.

[11] Yamakawa, T. and T. Miki, "The Current Mode Fuzzy Logic Integrated Circuits Fabricated by the Standard CMOS Process," *IEEE Transactions on Computers*, Vol. C-35, No. 2, pp. 161-167, February 1986.

[12] Yasunobu, S., S. Miyamoto, T. Takaoka, and H. Ohshima, "Application of Predictive Fuzzy Control to Automatic Train Operation Controller," *Proc. of IECON'84*, pp. 657-662, 1984.

[13] Zadeh, L. A., "Fuzzy set," *Information and Control,* Vol. 8 pp. 338-353, 1965.

[14] Zadeh, L. A., "Outline of a New Approach to the Analysis of Complex Systems and Decision-Making Approach," *IEEE Transactions on Systems, Man and Cybernetics,* Vol. SME-3, pp. No. 1, pp. 28-45, January 1973.

NOTES

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

S6-63
18362

N91-71356

p.20

Kaoru Hirota, Ph.D.

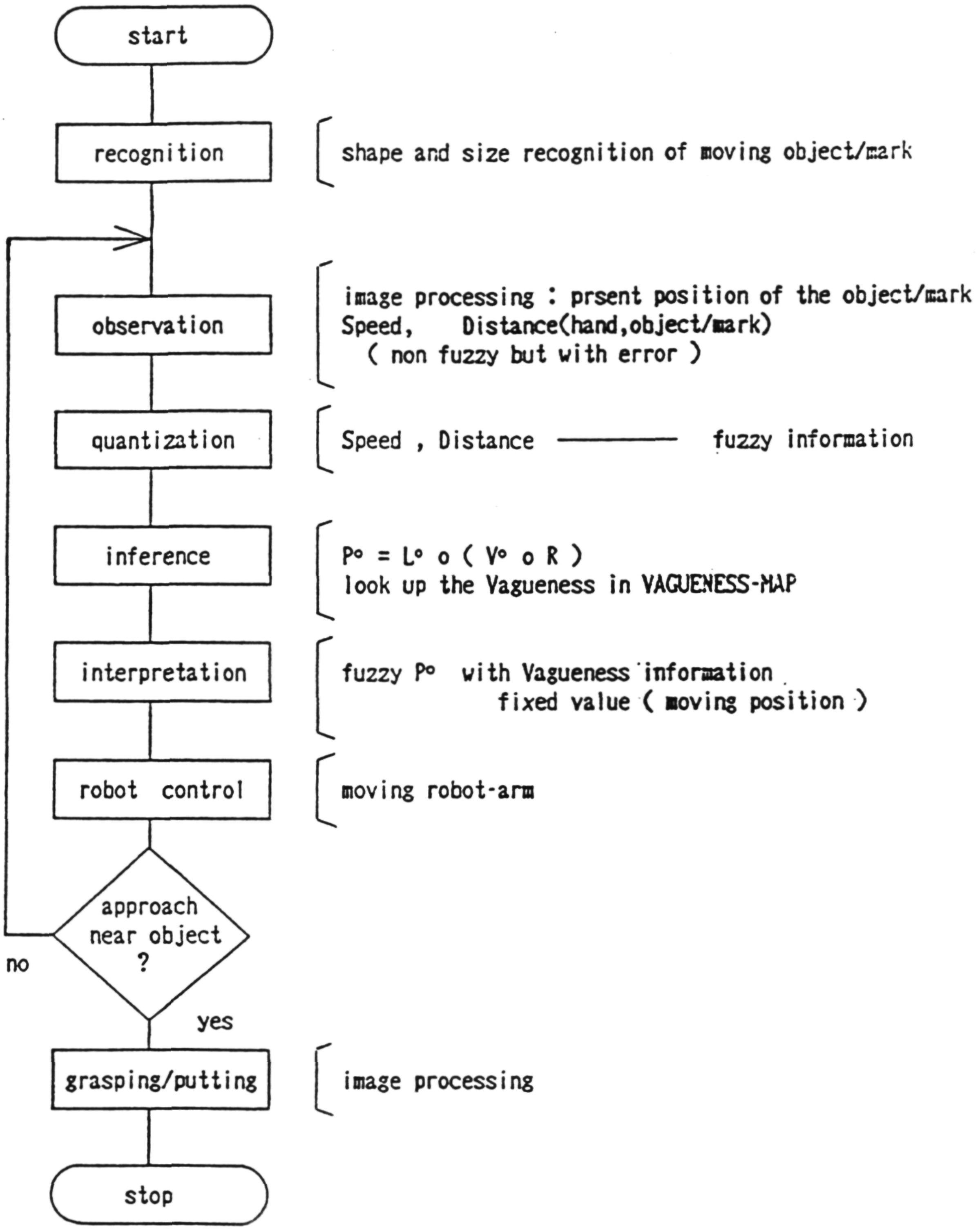
Hosei University
Tokyo, Japan

Dr. Hirota received his B.S. in electronics in 1974, his M.S. in electrical engineering in 1976, and his Ph.D. degree in electrical engineering in 1979 - all from the Tokyo Institute of Technology. From April 1979 to March 1982, Dr. Hirota was an assistant professor in the Department of Computer Sciences, Sagami Institute of Technology in Fujisawa, Kanagawa, Japan. He was an assistant professor from April 1982 to March 1983, and has been an associate professor since April 1983 in the Department of Instrument and Control Engineering, College of Engineering, Hosei University, Koganei, Tokyo, Japan. Dr. Hirota has been a part-time lecturer at the Sagami Institute of Technology since April 1982, Tokai University since April 1984, and the Technical College of FACOM since September 1986. Research interests include image pattern recognition, intelligent robotics, fuzzy control, artificial intelligence, and industrial applications of these subjects.

AN APPLICATION OF FUZZY LOGIC TO ROBOTIC VISION AND CONTROL

Abstract

A robot arm system able to manipulate a moving object on a belt conveyor at various speeds is built, consisting of two parts. The first part is related to recognizing patterns in real time. In this part, a method of constructing a fuzzy discriminant tree is proposed, where three newly defined measures called effectiveness, importance, and applicability are introduced. The robot arm system is able to recognize the shape and the size of moving patterns on a belt conveyor based on the fuzzy discriminant tree. The second part is to replace (grasp and put) a moving object based on fuzzy inference (or approximate reasoning) rules with the aid of an image processing technique. The whole system is controlled by one 16-bit personal computer and works in real time. The advantages of the proposed method are the reduction of processing time and the availability of low-level devices which have not been realized by other methods.



A flow chart of robot-arm system

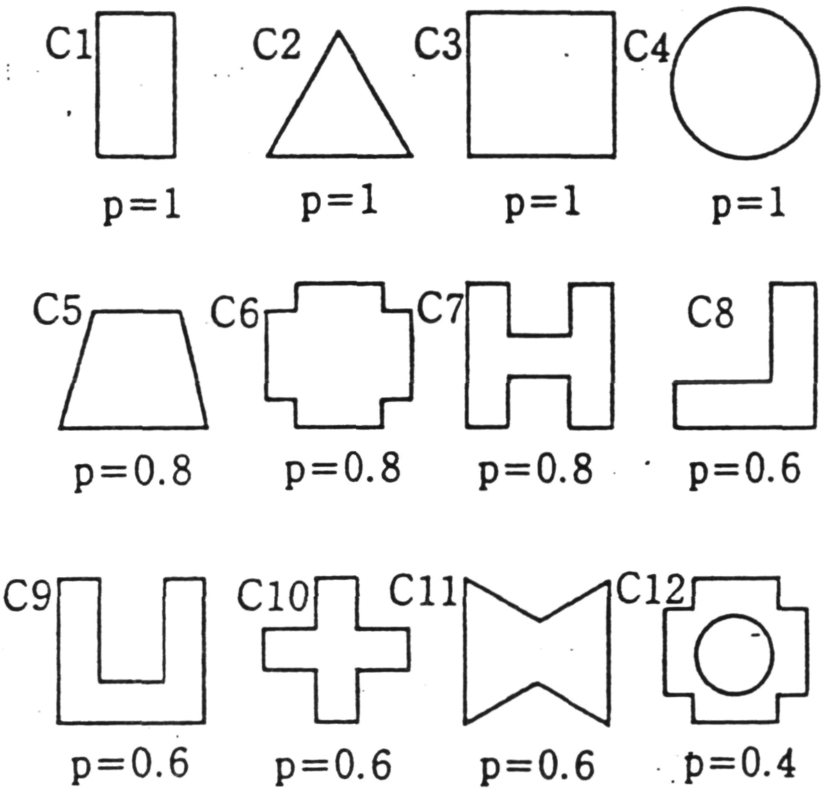


Fig. 6. Twelve patterns used in the shape recognition experiment.

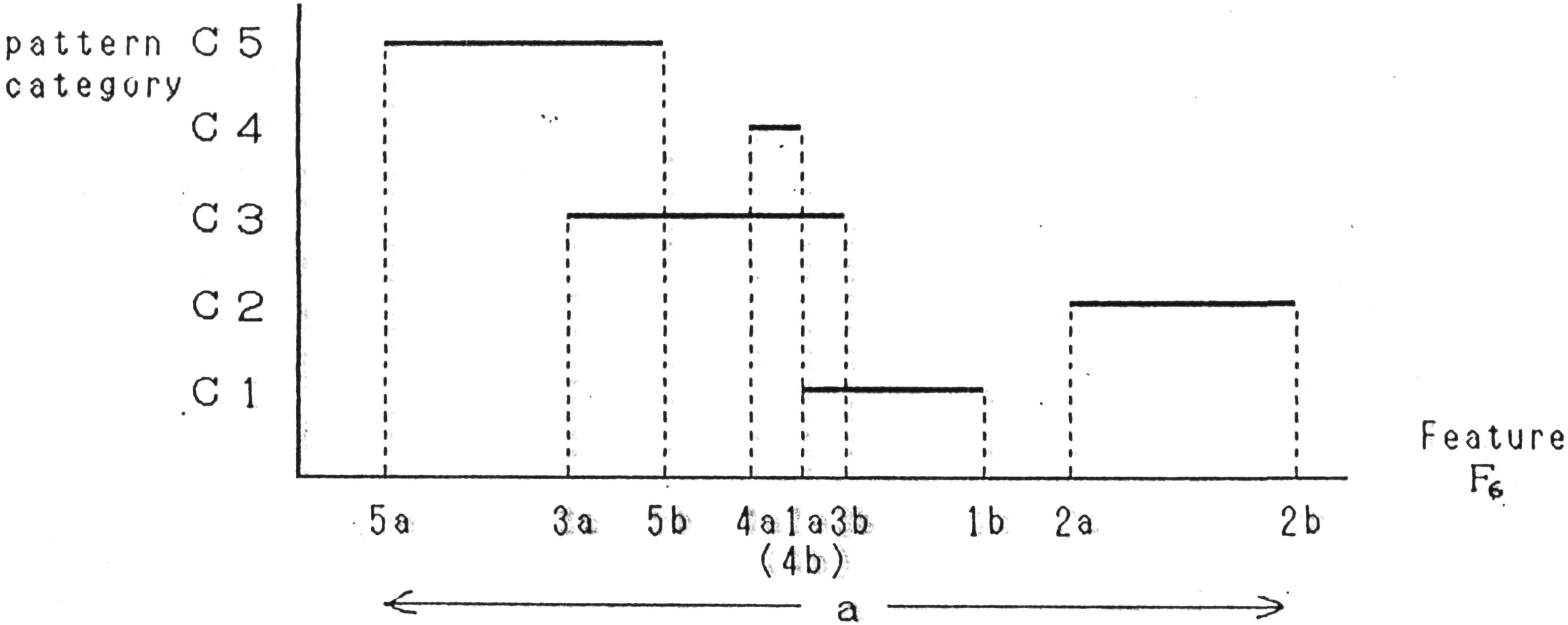
given features and their computing time in recognition process

	F_1 aspect ratio	F_2 variance of marginal distribution on x-axis	F_3 variance of marginal distribution on y-axis
computing time [sec]	0.25	0.808	0.807

	F_4 x-mean : max length	F_5 y-mean : max length	F_6 area density
computing time [sec]	0.778	0.60	0.59

	F_7 circum-area ratio	F_8 CG offset in x-axis direction	F_9 CG offset in y-axis direction
computing time [sec]	1.04	0.59	0.59

Computer PC-9800 (5MHz clock)
Language Assembler



Distribution map of Feature F_6

	C5	C4	C3	C2	C1
C1	○	×	×	○	
C2	○	○	○		
C3	×	×			
C4	○				
C5					

Discriminant table of Feature F_6

	C5	C4	C3	C2	C1
C1	0.15	×	×	0.1	
C2	0.45	0.3	0.25		
C3	×	×			
C4	0.1				
C5					

$E_6 = 1.35$

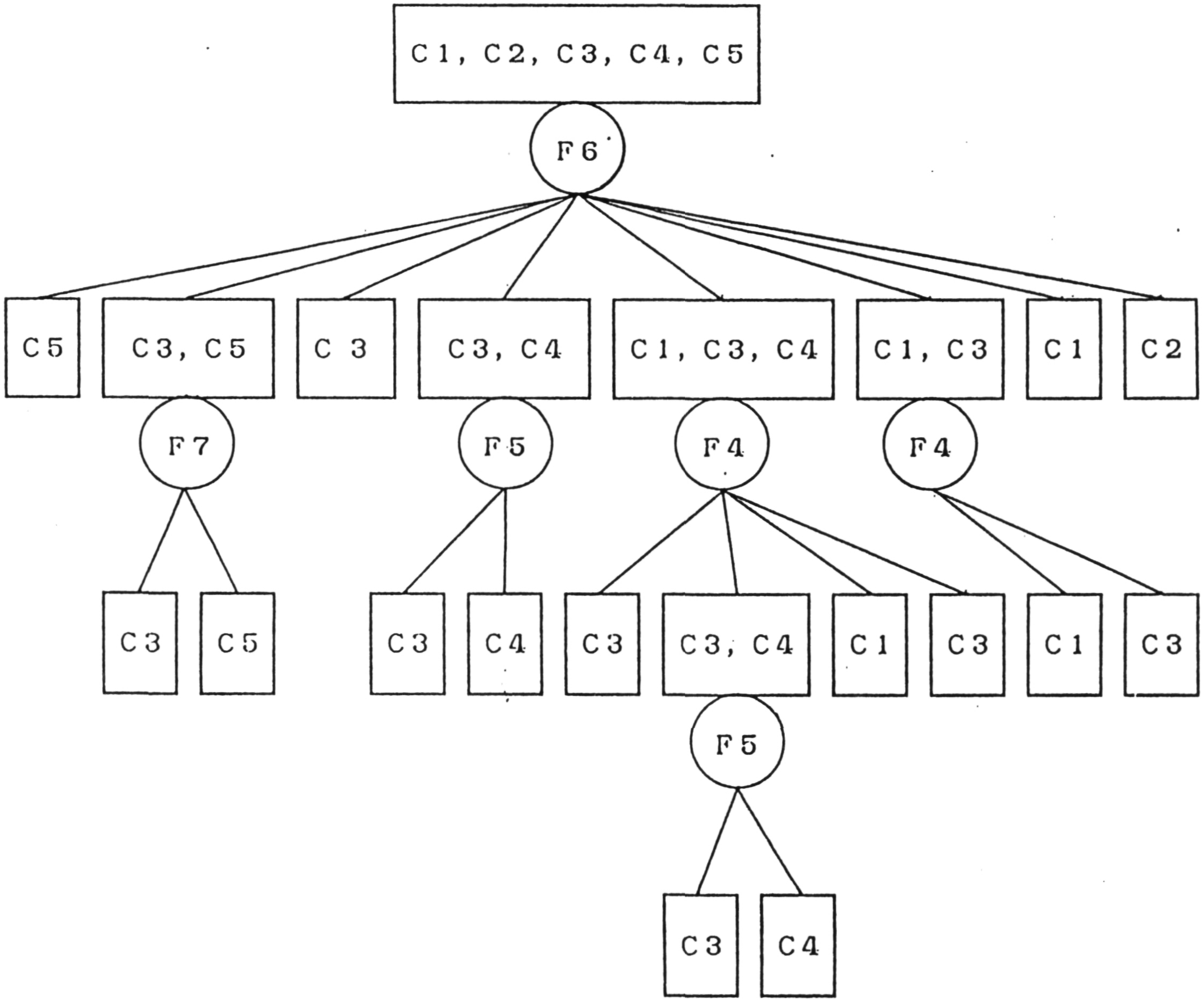
Effectiveness of Feature F_6

p_i		1	2	3	4	5
p_j		C5	C4	C3	C2	C1
5	C1	0.9	×	×	0.9	
4	C2	2.25	1.8	1.75		
3	C3	×	×			
2	C4	0.3				
1	C5					

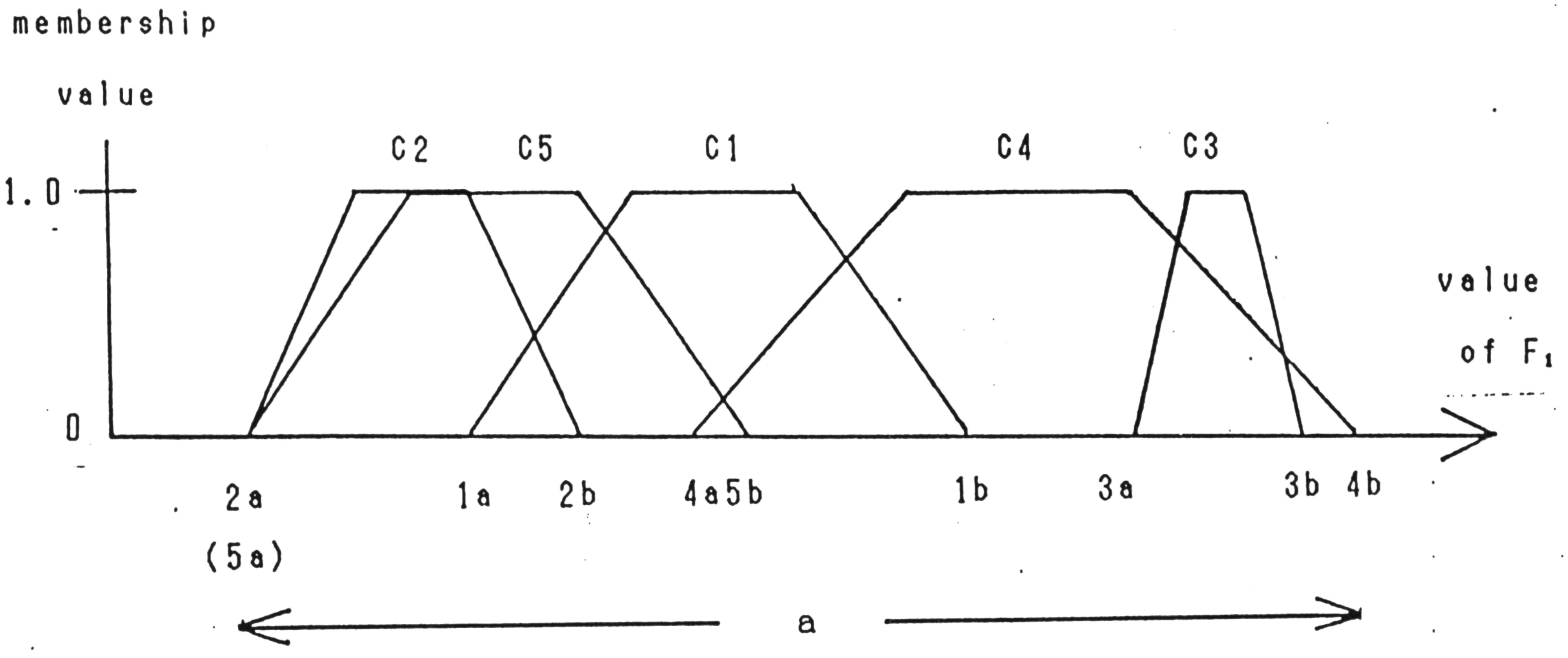
$\Sigma = 7.9 \quad t_6 = 2$

$I_6 = 3.95$

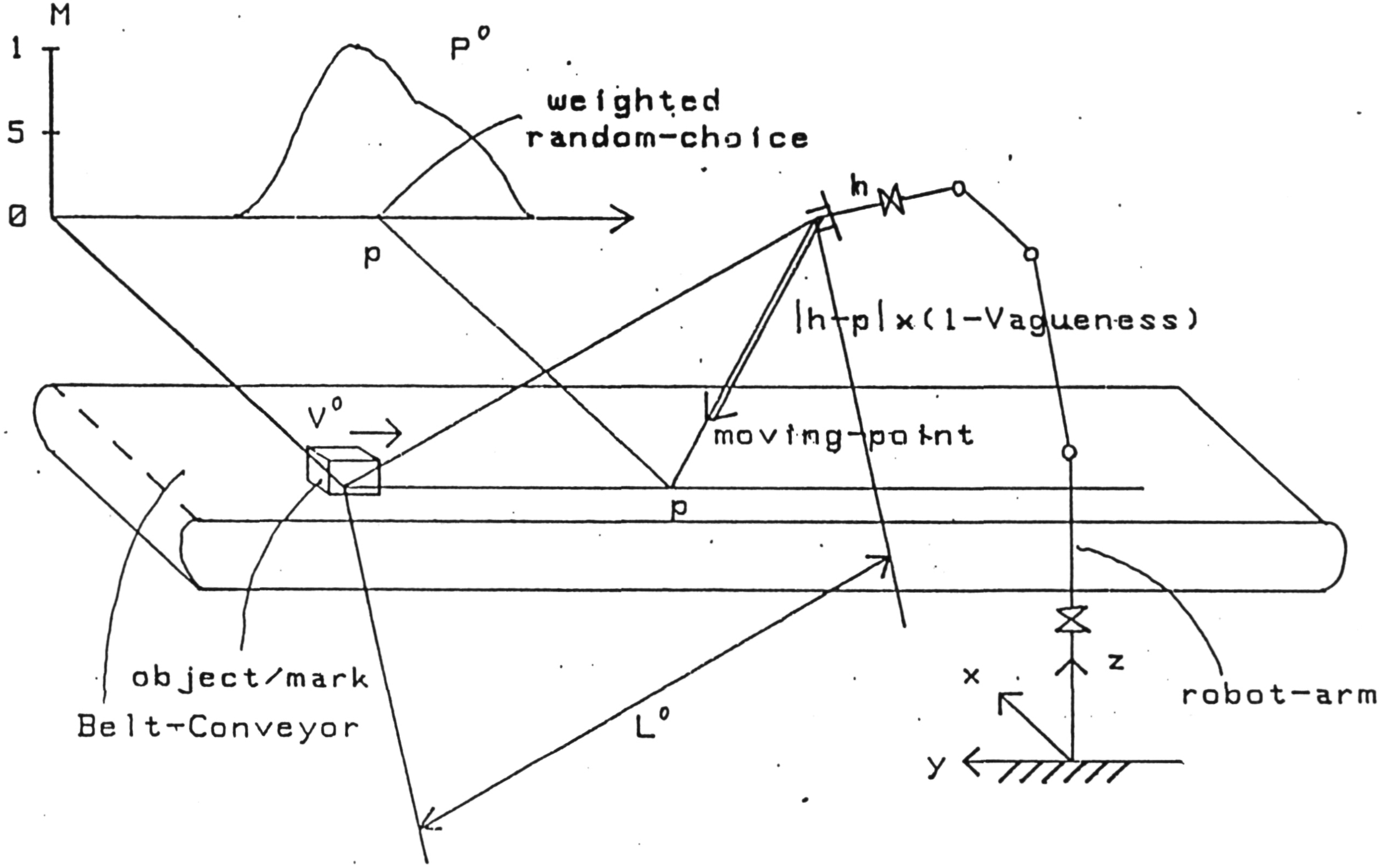
Importance of Feature F_6



An example of discriminant tree



distribution map of F_1



M
1
5
0
P°
weighted
random-choice
p
h
|h-p|x(1-Vagueness)
V°
moving-point
p
object/mark
Belt-Conveyor
L°
x
z
y
robot-arm

(e.g. If speed is "high" and distance is "far"
then move hand "far away")

```
IF V IS V1 AND L IS L1 THEN P IS P1  ELSE
IF V IS V1 AND L IS L2 THEN P IS P1  ELSE

IF V IS V4 AND L IS L5 THEN P IS P7  ELSE
IF V IS V4 AND L IS L6 THEN P IS P8
```

Vi: Speed Lj: Distance Pk: (estimated) Distance
i=1～4 j=1～6 k=1～8

Rule map

near <-----------------> far

		L1	L2	L3	L4	L5	L6
low	V1	P1	P1	P1	P1	P1	P1
	V2	P1	P2	P2	P3	P4	P5
	V3	P1	P2	P4	P5	P6	P7
high	V4	P1	P3	P5	P6	P7	P8

a little a way <-----------> far a way

P1 P2 P3 P4 P5 P6 P7 P8

Vagueness map

	L1	L2	L3	L4	L5	L6
V1	0	0	0	0	0	0
V2	0	0	0	.2	.3	.4
V3	0	.2	.2	.3	.4	.5
V4	0	.3	.3	.4	.5	.5

Fuzzy labels of V , L , P

(a) Fuzzy labels of Speed (V)

low <-----------------------------------> high

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
V1	1	1	1	1	.5	.1	0	0	0	0	0	0	0	0	0
V2	0	.2	.4	.8	.1	.8	.4	.2	0	0	0	0	0	0	0
V3	0	0	0	0	0	.2	.4	.8	1	.8	.4	.2	0	0	0
V4	0	0	0	0	0	0	0	0	0	.2	.4	.8	1	1	1

(b) Fuzzy labels of Distance between object and robot-hand (L)

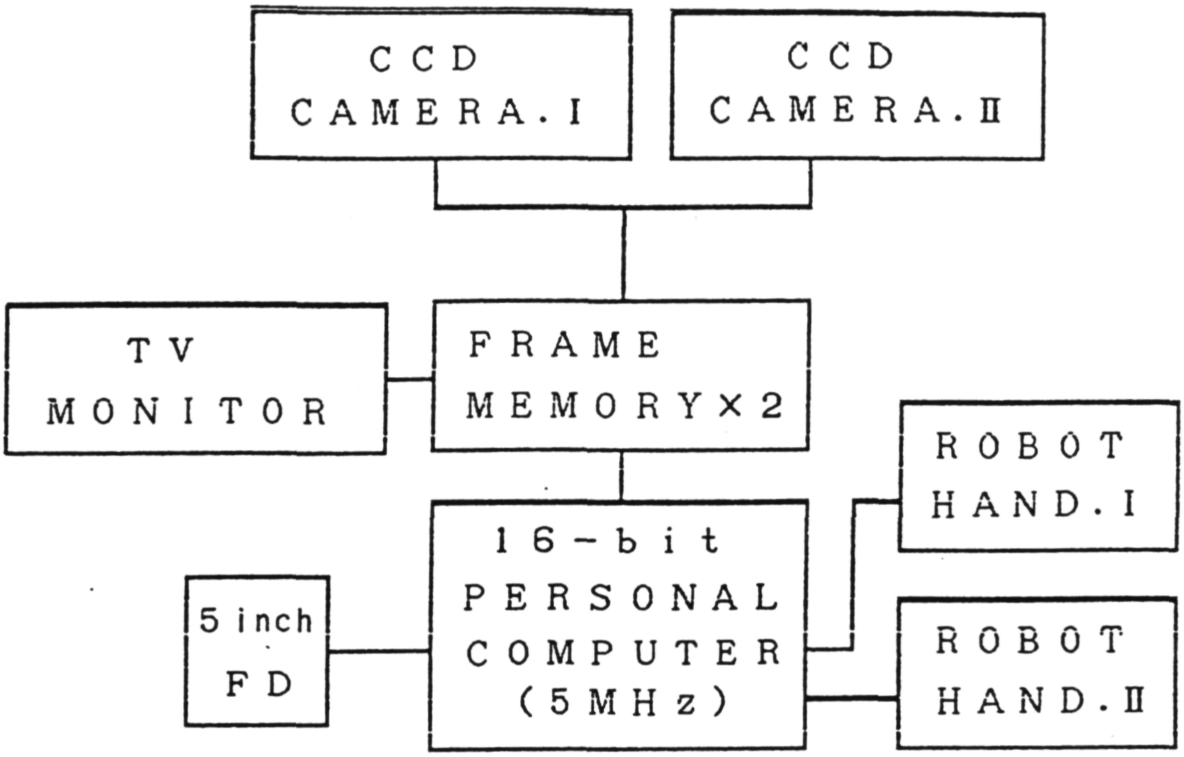
near <---------------------------------> far

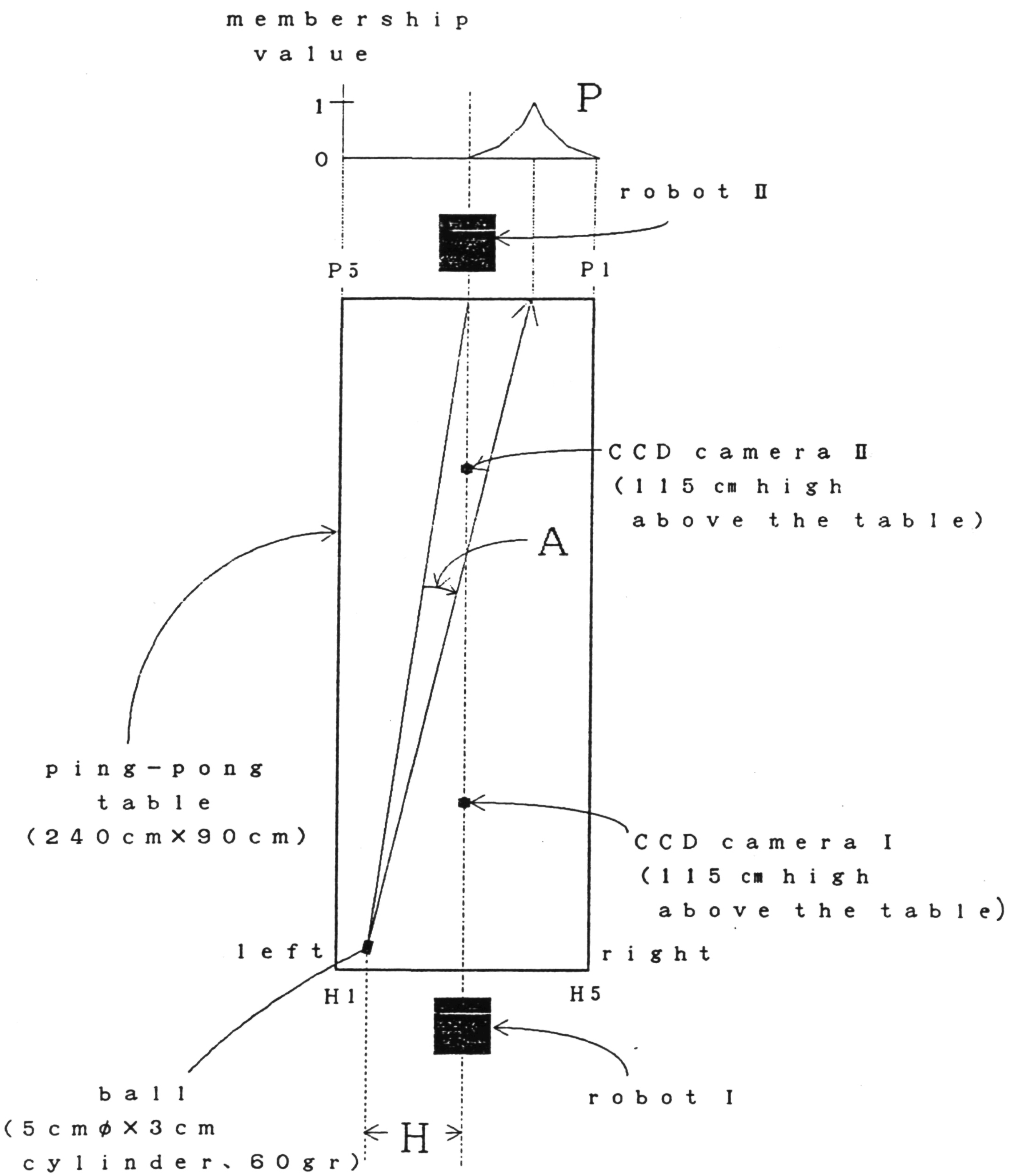
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
L1	1	.8	.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L2	.1	.6	1	.6	.1	0	0	0	0	0	0	0	0	0	0	0	0
L3	0	0	.1	.6	1	.6	.1	0	0	0	0	0	0	0	0	0	0
L4	0	0	0	0	.1	.6	1	1	1	.6	.1	0	0	0	0	0	0
L5	0	0	0	0	0	0	0	0	.1	.6	1	1	1	.6	.1	0	0
L6	0	0	0	0	0	0	0	0	0	0	0	0	.1	.6	1	1	1

(c) Fuzzy labels of (estimated)moving-Distance (P)

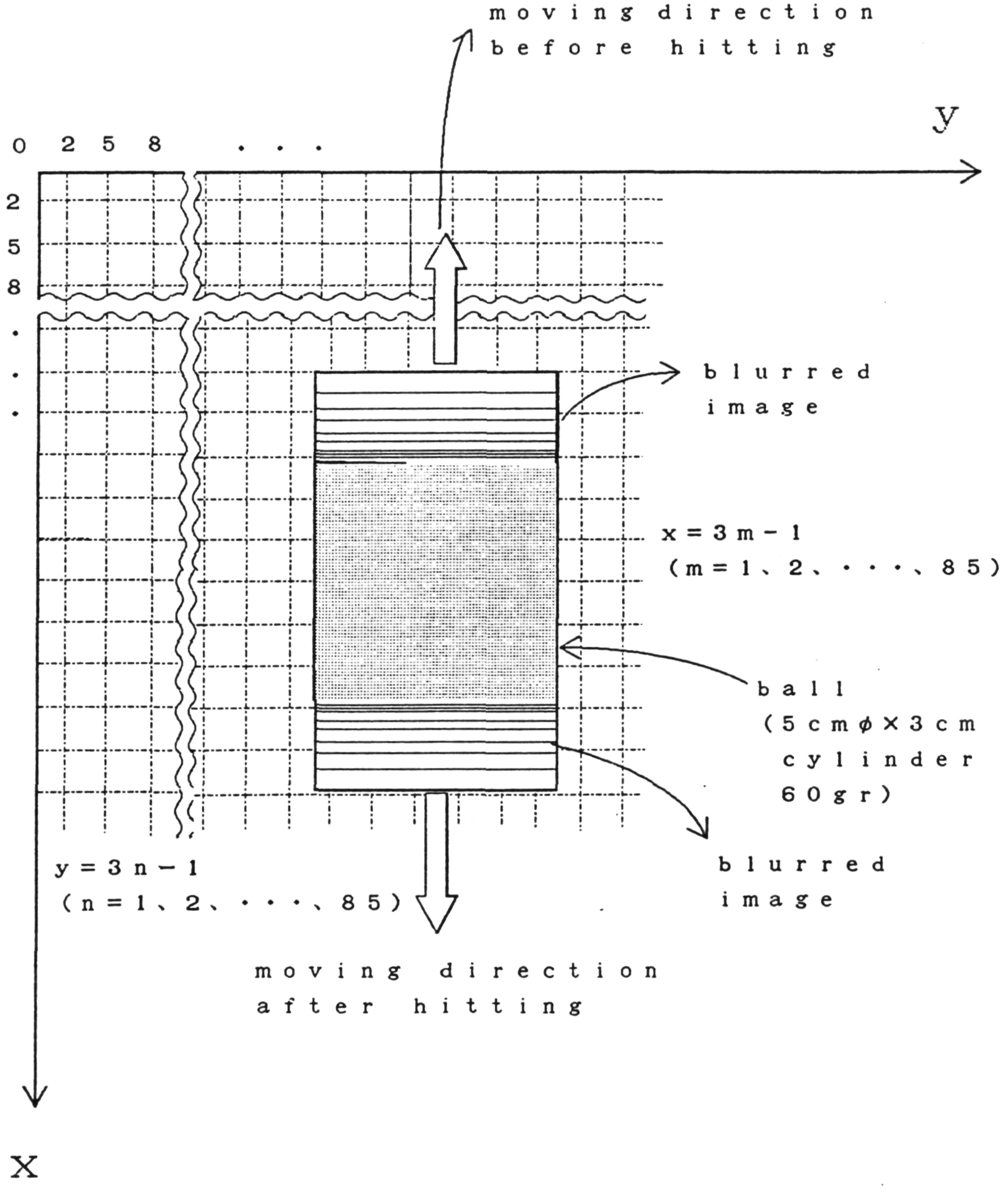
a little away <-----------------> far a way

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P2	.1	.6	1	.6	.1	0	0	0	0	0	0	0	0	0	0	0	0
P3	0	.2	1	.2	0	0	0	0	0	0	0	0	0	0	0	0	0
P4	0	0	.1	.6	1	.6	.1	0	0	0	0	0	0	0	0	0	0
P5	0	0	0	0	.1	.6	1	.1	.6	.1	0	0	0	0	0	0	0
P6	0	0	0	0	0	0	.1	.6	1	.6	.1	0	0	0	0	0	0
P7	0	0	0	0	0	0	0	0	.1	.6	1	1	.6	.1	0	0	0
P8	0	0	0	0	0	0	0	0	0	0	0	.1	.6	1	1	1	1


CCD CAMERA.I
CCD CAMERA.II
TV MONITOR
FRAME MEMORY×2
ROBOT HAND.I
16-bit PERSONAL COMPUTER (5MHz)
5 inch FD
ROBOT HAND.II



membership value
1
0
P
robot Ⅱ
P 5
P 1
CCD camera Ⅱ (115 cm high above the table)
A
ping-pong table (240cm×90cm)
CCD camera Ⅰ (115 cm high above the table)
left
right
H 1
H 5
robot Ⅰ
ball (5cmϕ×3cm cylinder、60gr)
H


moving direction
before hitting
y
0 2 5 8 ・・・
2
5
8
・
・
・
blurred
image
x = 3 m − 1
(m = 1、2、・・・、85)
ball
(5cmφ×3cm
cylinder
60gr)
blurred
image
y = 3 n − 1
(n = 1、2、・・・、85)
moving direction
after hitting
x

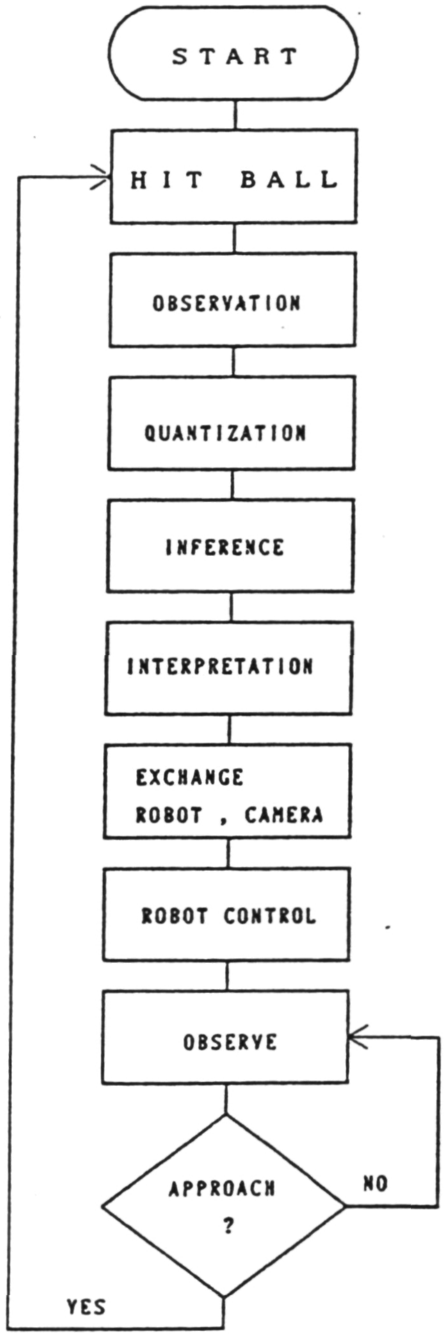

START
HIT BALL
OBSERVATION
QUANTIZATION
INFERENCE
INTERPRETATION
EXCHANGE
ROBOT , CAMERA
ROBOT CONTROL
OBSERVE
APPROACH
?
NO
YES

Table 1 Fuzzy labels of H, A, P

(a) Fuzzy labels of H

			left		--										right				
	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9
P1	1	1	.9	.6	.2	.1	0	0	0	0	0	0	0	0	0	0	0	0	0
P2	0	.1	.2	.6	.9	1	.9	.6	.2	.1	0	0	0	0	0	0	0	0	0
P3	0	0	0	0	0	.1	.2	.6	.9	1	.9	.6	.2	.1	0	0	0	0	0
P4	0	0	0	0	0	0	0	0	0	.1	.2	.6	.9	1	.9	.6	.2	.1	0
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	.1	.2	.6	.9	1	1

(b) Fuzzy labels of A

			negative		--												positive				
	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
A1	1	1	.9	.6	.2	.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	0	.1	.2	.4	.8	1	1	.8	.4	.2	.1	0	0	0	0	0	0	0	0	0	0
A3	0	0	0	0	0	0	.1	.2	.6	.9	1	.9	.6	.2	.1	0	0	0	0	0	0
A4	0	0	0	0	0	0	0	0	0	0	.1	.2	.4	.8	1	1	.8	.6	.2	.1	0
A5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.1	.2	.6	.9	1	1

(c) Fuzzy labels of P

			left		--										right				
	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9
P1	1	1	.9	.6	.2	.1	0	0	0	0	0	0	0	0	0	0	0	0	0
P2	0	.1	.2	.6	.9	1	.9	.6	.2	.1	0	0	0	0	0	0	0	0	0
P3	0	0	0	0	0	.1	.2	.6	.9	1	.9	.6	.2	.1	0	0	0	0	0
P4	0	0	0	0	0	0	0	0	0	.1	.2	.6	.9	1	.9	.6	.2	.1	0
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	.1	.2	.6	.9	1	1

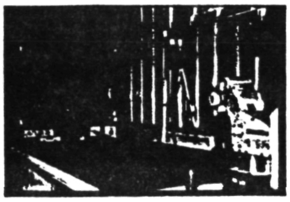
Tai

	order of fuzzy inference	1st	2nd	3rd	4th	5th	6th	7th
	corresponding photos in Photo 1 (a)	(2) (3)	(5) (6)	(8) (9)	(10) (11)	(12) (13)	(14) (15)	(16)
Antecedent	(H°) observed hitting posi (mm)	0	11	−296	−350	−4	314	387
	(A°) obsered hitting angl (°)	0	5. 8	7. 1	0	−6. 2	−7. 2	−0. 5
Conclusion	(P°) inferred moving posi (mm)	0	−285	−367	0	300	376	−30

(1)

(2)

(3)

(4)

(5)

(6)

(7)

(8)

(9)

(10)

(11)

(12)

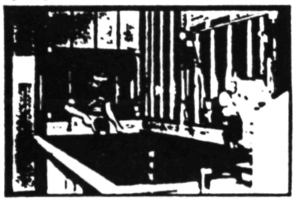
(13)

(14)

(15)

(16)


(1)

(2)

(3)

(4)

(5)

(6)

(7)

(8)

(9)

(10)


Photo 1

NOTES

57

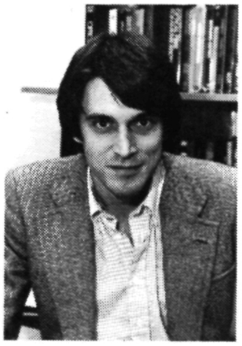
57-60

ABS. ONLY

15307

N91-71357

p.1



Demetri Psaltis, Ph.D.

California Institute of
Technology
Pasadena, California

Dr. Psaltis received his B.S. in electrical engineering and economics in 1974 and his M.Sc. and Ph.D. degrees in electrical engineering in 1975 and 1977, respectively, all from Carnegie-Mellon University, Pittsburgh, Pennsylvania. Upon completion of the Ph.D. program, he remained at Carnegie-Mellon as a research associate, and later as a visiting assistant professor, for a period of 3 years. In 1980, Dr. Psaltis joined the faculty of the electrical engineering department at the California Institute of Technology, Pasadena, where he is now an associate professor and consultant to industry. His research interests are in the areas of optical information processing, acoustooptics, image processing, pattern recognition, neural network models of computation, and optical devices. He has written more than 130 technical publications, is a fellow of the Optical Society of America, and is vice president of the International Neural Networks Society.

OPTICAL NEURAL COMPUTERS

Abstract

A neural computer consists of a large number of simple processing elements (neurons) that are densely interconnected. The information that is needed to solve a particular problem is stored in the strength of the interconnections using a learning procedure. Some of the basic characteristics of such a computer and the class of problems for which it is best suited are discussed. Optics is a technology particularly well suited for implementing neural computers because of the relative ease with which a programmable, massive interconnection network can be optically synthesized. Several experimental demonstrations of optical networks will be described and the ultimate capabilities of optical neural computers will be projected.

NOTES

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

S8-62
153088
N91-71358
P.20

Harold Szu, Ph.D.

Naval Research
Laboratory
Washington, D.C.

Dr. Szu received his Ph.D. from Rockefeller University in 1971, and from 1972 to 1974, he taught at the University of North Carolina at Chapel Hill. He served 5 years in the Plasma Physics Division of NRL, and obtained two patents for (1) an infrared free-electron laser, and (2) a superconducting heavy-ion accelerator. In 5 years at the Optical Sciences Division of NRL, Dr. Szu obtained two patents for (1) a phase-conjugate-crystal interferometer, and (2) a computer-generated-hologram pattern classifier. Two patents (pending) are a result of 3 years in Tactical Electronic Warfare at NRL, (1) a nonconvex optimization Cauchy machine, and (2) superconducting sensor arrays, neuromorphic computers, and triode amplifiers. In addition to more than 100 technical papers and reports, Dr. Szu has published three books: "Fluctuation-Dissipation Theorems," "Optical and Hybrid Computing," and "Pattern Recognition Based on Human Visual Systems and Neural Networks" for Lawrence-Erlarbaum. Dr. Szu also teaches Adaptive Neural Networks at UCLA, serves as a governing board member and officer of the International Neural Network Society (INNS), and is the editor of Pergamon Journal, Neural Networks.

WHAT IS THE SIGNIFICANCE OF NEURAL NETWORKS FOR AI?

Abstract

Associative memory (AM) and attentive associative memory (AAM) have been reviewed in terms of simple neural networks (both uniform and nonuniform matched filter banks - read by inner products and written by outer products in parallel). Whereas AM has been applied to optical character recognition (OCR) using the set of orthogonal feature vectors deduced from image processing and computer vision, AAM can incorporate AI expert system techniques for determining the nonuniform linear combination of outer products. A rule-based system can more efficiently incorporate the frequency distribution of distorted characters according to user group profiles; i.e., left-handed versus right-handed writing. Specifically, in this paper we have examined the degree of fault tolerance in AM, the ability of generalization by interpolation (auto-associative memory), and abstraction by extrapolation (hetero-associative memory). The efficiency of the closed system of rule-based knowledge representation of AI using tuple storage has been combined with the flexibility of the non-rule-based open system using the matrix knowledge representation of NI (coined for either neural, or network, or natural intelligence). Thus, the ability of generalization and abstraction becomes possible in a combined intelligent system of AI and NI.

WHAT IS THE SIGNIFICANCE OF NEURAL NETWORKS FOR AI ?

by Harold H. Szu
Naval Research Laboratory, Code 5756
Washington D.C. 20375

ABSTRACT

Associative memory (AM) and attentive associative memory (AAM) have been reviewed in terms of simple neural networks (both uniform and non-uniform matched filter banks: read by inner products and write by outer products in parallel). While AM has been applied to the optical character recognition (OCR) using the set of orthogonal feature vectors deduced from image processing and computer vision, AAM can incorporate AI expert system techniques for determining the non-uniform linear combination of outer products. A rule-based system can more efficiently incorporate the frequency distribution of distorted characters according to user group profiles, say left-handed writing versus right-handed writing. Specifically in this paper, we have examined the degree of fault tolerance in AM, the ability of generalization by interpolation (auto-associative memory) and abstraction by extrapolation (hetero-associative memory). The efficiency of the closed system of rule-based knowledge representation of AI using the tuple storage has been combined with the flexibility of the non-rule based open system using the matrix knowledge representation of NI (coined for either Neural, or Network, or Natural Intelligence). Thus, the ability of generalization and abstraction becomes possible in a combined intelligent system of AI and NI.

1. INTRODUCTION

The question of the significance of neural networks for AI may be subdivided into three aspects.

(i) How can neural networks help solve AI problems ?

ANSWER: Both the well understood fault-tolerance of associative memory (AM), and the lesser understood ability of neural networks for generalization and abstraction, can be usefully incorporated into AI techniques.

(ii) How can AI help solve neural network problems ?

ANSWER: Similar to computer aided design, AI expert systems with a neural network modules can help design special purpose architectures for neural network computing.

(iii) What unsolved problems can be solved efficiently by combining AI and NI (coined for either Neural, or Network, or Natural Intelligence) techniques to utilize their respective strengths?

ANSWER: The optical character recogniton (OCR) for reading hand-written bank check and zip-codes, can be solved by combining both AI and NI techniques, as described in this paper.

Because we can only build a small neural network, we wish to endow a small se neurons with a human-like intelligence. With present technology, whether it be electronic optical, one cannot build a neural network of more than several hundred neurons, using existi processor elements (PE's), because of the technological difficulty associated with der interconnectivity, about N^2 for N PE's. Thus, artificial neural networks can not yet match the si and the complexity of the human brain, that has billions of neurons and thousands interconnects for each neuron. If we are ***not***, overly ambitious in developing a **general purpo** neural computer, we can built a **special purpose** neural computer for solving special purpo problems, such as OCR.

One way to accomplish this special purpose neural computer is to combine the tradition rule-based AI wisdom with non-rule-based NI learning. This is particularly desirable in solvi OCR problems because the available small neural networks can use better feature vectors obtain from other disciplines. Neural networks, built with current technology, can then provide fau tolerance for input feature vectors variations. The specific problem of hand-written charact recognition, differs from the more regular, hand-printed, alphanumeric recognition problem that it must account for such complications as connected characters and characters broken segmentation.

Conceptually, one could solve the OCR problem using analytic, rule-based AI or neur network techniques. The OCR problem can be subdivided into character (or character strin statistics, font recognition, and character recognition; the most efficient techniques for these thr subproblems are analytic (statistical), rule-based AI, and neural networks, respectively. Since t statistical techniques, applied to alphanumeric frequencies, is well known, this topic will not discussed further. In solving the font recognition subproblem, AI rules can be set by the (statisti frequency distribution of individual distorted characters according to user group profiles, e.g. le handed writing versus right-handed writing. It is efficient to design AI expert system that dra upon the classical statistical pattern recognition, e.g. one stroke difference exists between "*P* " a "*R* ", or between"*O* " and "*Q* ", or in a low pass filter viewpoint only one stroke locatio difference exists among four rounded letters "*P* " and "*R* ","*O* ", and "*Q* ". Furthermore, the rules of **pair character distortion distribution** can help solve the problem of ***connected charac*** and ***broken character after segmentation***. such as **two scripted zeros**. The **pair characer correlati matrix can be analyzed by the technique of the Karhunen-Loeve** procedure in image processi The Karhunen-Loeve technique is compatable with AM's outer product decomposition. With t help of AI rule-based system, both the first and the second order statistics can be incorporated in t formalism of **attentive associative memory** (AAM), that processess the extra degrees of freedom the non-uniform storage of vector outer products based on a given set of critical feature vectors.

Because the open-ended knowledge of input pattern variations may be efficien controlled by using other disciplinary knowledge, such as AI and computer vision with a result better combined technology, we shall review AM and AAM, and various OCR approaches means of their specific techniques used for feature extraction and techniques used for gro classification. The sooner we accept implementation limitations of the present neurocompute the better we can work with other disciplinary researchers. For example, we can work w researchers in AI, computer vision, image processing. Since this cross disciplinary collaboration

by nature not easy because of different trainings and languages involved, then this paper may serve a door opener for both.

Pattern recognition reseachers have been successful in machine-printed character recognition (CR) compared to optical character recognition (OCR) of hand-written bank checks or zipcodes. Difficulties of applying AI alone to an intelligent OCR may be due to the lack of non-rule-based capability of generalization and abstraction. This may be constrained by the traditional AI **one dimensional (1-D) knowledge representation**, e.g. an ordered set of tuples used in semantic networks. Similarly, difficulties of applying the neural network alone to an intelligent OCR may be in selecting critical features that is precisely one of the most challenging and unsolved problems (others are segmentations and locations). On the other hand, AI is efficient in reduce the problem to a sub-problem based on **1-D** knowledge representation of simple rules, and NI provides the fault-tolerant OCR system based on **2-D** knowlege representation. Together they give the possibility of generalization and abstraction. Thus, Szu and Tan (1988) have considered a less risky approach that consists of the traditional AI researchers who know about OCR critical features, and the neural network experts who know about AM fault tolerance. Technological developments have pointed to the readiness of such collaborations, since 2-D storage by chips or optical disks becomes cheaper than the traditional 1-D content addressable memory (CAD) processor. What's needed is a smart coprocessor such as neurocomputer. As a matter of fact, due to the 2-D nature of light, optical expert systems based on AM have been designed by Szu and Caulfield (1987) who have shown as simple replacement of 1-D tuples by 2-D matrices in a semantic network the alias problem for data fusion is solved by matrix addition and thresholding. The opto-electronical implementation of attentative associative memory model of Athale, Szu & Frielander (1986) can be expanded by means of a priori probability compiled by a pair-character correlation function of script letters. These papers may facilitate both sides the starting line of collaborations.

In this paper, we have reviewed the orthogonal subspaces of features and examined (1) the degree of fault tolerance , (2) the generalization by interpolation to other orthogonal feature vectors within the subspace, and (3) the abstraction by extrapolation to other subspaces. **AAM** may be formulated by a linear combination of outer products based on a set of orthogonal feature vectors. The combination coefficient is called the attention parameter, because it enters into the eigenvalue of **AAM** matrix that governs the recall convergence. We review briefly about the dynamics of attentive associative memory published by Szu (1988) elsewhere using arbitrary coefficients. In this paper we explicitly introduce a AI-tuple for the attention vector $\mathbf{a} = \{a_n , n=1,...M\}$, where the inner product between the difference vector between an averaged stochastic input $|\mathbf{Q}>$ and a fixed memory state $|\mathbf{m}>$ is naturally used as the attention parameter defined in terms of Dirac's inner product notation: $a_m = <m|m> - <m|Q>$. Such an **AAM matrix** has non-white eigenvalue spectrum $\lambda_n \equiv a_n - (A / B)$ where the attentive memory capacity is $A \equiv \Sigma^M_{n=1} a_n$, and B is the length of the feature vectors (e.g. the number of bits). Iterative recalls are used. **Paying non-uniform attention** ($a_n \geq 1$) **increases the memory capacity** $A \geq M$ **together with a faster convergence rate proportional to the larger eigenvalue** $\lambda_m \geq \lambda$ **than a uniform attention**(i.e.a_m = 1). Szu's (1988) analysis has suggested that the eigenvalue spectrum and its dithering by input ensemble can play a crucial role for the convergence associated with a nonlinear dynamical system.

2. Associative Memory

Matrix associative memory works like a parallel bank of matched filters but much m efficiently in at least three counts: (1) no address coding of input and decoding for output necessary , (2) operations are done in parallel, and (3) the connectivity matrix can be determi by itself using various adaptive (learning) algorithms.

An analytical and numerical example of AM is given as follows:

We denote M feature vectors as binary words, $U^{(m)}$, m=1,...M. Each word has B bits. inner product of Eq(1) measures the norm, the number of bits that are one.

$$U^{T} \bullet U = \#\ \text{of one's} \tag{1}$$

where the superscript transpose the column vector to a row vector.

The associated **bipolar words**, denoted by $V^{(m)}$, m=1, ...M, are defined as follows:

$$V = (2\,U - 1) = \mathrm{Sgn}(\,U\,) \tag{2}$$

where the unit vector **1** has all entries equal 1 and Sgn is the sign function that changes zero negative quantities to -1. We prefer bipolar version to binary version because : (1) the in product norm is always identical to the number of bits, B:

$$V^{T} \bullet V = B = \langle V \mid V \rangle\ , \tag{3}$$

rewritten here in terms of Dirac's bracket notation: $\langle bra \mid ket \rangle$ for the inner and $\mid ket \rangle \langle bra \mid$ for outer product, (2) the nature of "exclusive or" can be easily represented by bipolar multiplication

$$+1 \times +1 = 1,\ -1 \times -1 = 1,\ +1 \times -1 = -1,\ -1 \times +1 = -1,$$

(3) the inner product norm is related to the Hamming distance, defined to be the numbe different bits between two vectors no matter where the differences occur.

We assume an orthogonal set of feature vectors defined as follows:

$$V^{(n)T} \bullet V^{(m)} = B\ \delta_{n,m} = \langle\, \mathbf{n} \mid \mathbf{m} \,\rangle \tag{4}$$

where $\delta_{n,m}$ is the Kronecker delta. The outer product weight matrix **W** represents au associative memory:

$$[\,W\,] = \Sigma_m\,[\,V^{(m)} V^{(m)}] = \Sigma_m \mid \mathbf{m} \rangle \langle \mathbf{m} \mid \tag{5}$$

Hopfield (1982, 1984) assumed the auto-associative matrix [T] to be traceless. That was used together with the symmetry property to prove convergence. Thus, the second term of Kronecker's delta matrix (1's along the main diagonal and zero elsewhere) is introduced in Eq (6) to make it traceless.

$$B[T]_{ij} = [W]_{ij} - M\,\delta_{i,j} \qquad (6)$$

B is the normalization constant, and M is the memory capacity. Using the trace operation denoted by Tr, we can easily verify Eq (6) to be traceless.

$$\mathrm{Tr}(\,|m\rangle\langle m|\,) = B \qquad (7)$$

$$\mathrm{Tr}(\,[\delta_{i,j}]\,) = B \qquad (8)$$

The tradeoff between the memory capacity and the degree of fault-tolerance has been estimated to be about 15 % of B bits [Hopfield (1982)] for pseudo-orthogonal vectors. That is,

$$M = 0.15\,B \qquad (9)$$

For orthogonal feature vectors, however, the capacity is 100 %.

$$M = B \qquad (10)$$

This fact can be demonstrated by the eigenvalue problem of the matrix which is defined to be

$$[T]\,|n\rangle = \lambda_n\,|n\rangle \qquad (11)$$

where the eigenvalue can be easily verified, using Eqs (4) and (6), to be *degenerate* , namely, a white spectrum for all M states,

$$\lambda_n = 1 - (M/B) \qquad (12)$$

The full capacity, M = B, corresponds to a zero eigenvalue for all B orthogonal eigenstates, one for each feature vector.

Consider a simple example where B = 4. There are 4 possible orthogonal vectors and 2^4 = 16 possible words denoted by:

0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15

We introduce orthogonal subspaces defined by the number of contiguous 1's in the binary word. The subspace consisting of words 13, 11, 7, and 14 is obviously orthogonal by shifting a "one" among 3 zeroes from the the left to the right end of the word.

Word	Binary	Word		Binary		p	Word	Bipolar	Word Bipolar
	p		comple.					comple.	
13	1101	2	0010	3	13	1 1 -1 1	2	-1-1 +1-1	3
11	1 011	4	0100	3	11	1 -1 1 1	4	-1+1-1 -1	3
7	0111	8	1000	3	7	-1 1 1 1	8	+1-1-1-1	3
14	1110	1	0001	3	14	1 1 1 -1	1	-1-1-1+1	3
15	1111	0	0000	4	15	1 1 1 1	0	-1-1 -1 -1	4
6	0110	9	1001	2	6	-1 1 1-1	9	+1-1-1+1	2
12	1100	3	0011	2	12	1 1 -1-1	3	-1-1+1+1	2
10	1010	5	0101	1	10	1-1 1-1	5	-1+1-1+1	1

It is readily verified that the subspace of bipolar words (13, 11, 7, 14) are mutua orthogonal to one another, as shown in Figure 1. They happen to be related to the Walsh functi of periodicity p=3. The corresponding binary words have an equal angle among them [$\cos^{-1}$ (2/ that is not 90^{o}. Also, the second subspace of bipolar words (15, 6, 12, 10) are also orthogonal but two subspaces are *not* orthogonal to each other.

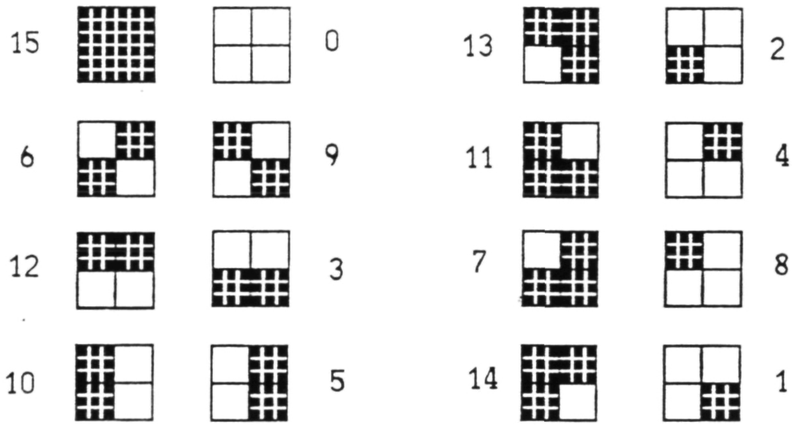


Figure 1. Two-Dimensional Representation of Walsh Base Functions Used to illustrate the fault tolerance and generalization properties of Associative Memory

We consider the storage of one word in memory.

$$4\,[\,T_1\,] = [13] = |13\rangle\langle 13| - \delta \tag{13}$$

If the outer product is properly normalized, it is related to the projection operator:

$$[P] = \delta - |13\rangle\langle 13| (1/B) \tag{14}$$

Using Eq (4) , it can be verified that

$$[P]^2 = [P]. \tag{15}$$

We will show (1) the ability of fault tolerance, and (2) the ability for generalization.

Fault Tolerance

The following sequence of erasing (zero out) successively from the bipolar bits illustrate tolerance of missing bits.

(1) one missing bit

$$[13](0\ 1\ {-1}\ 1)^T = \mathrm{Sgn}(3\ {-2}\ {-2}\ 3)^T = |13\rangle \tag{16}$$

where Sgn is sign function representing the sigmoid neuron response by the point nonlinearity extracting the algebra sign of each entries.

(2) two missing bits

$$[13](0\ 0\ {-1}\ 1)^T = \mathrm{Sgn}(2\ \ 2\ {-1}\ 1)^T = |13\rangle \tag{17}$$

(3) three missing bits

$$[13]\ (0\ 0\ 0\ \ 1)^T = \mathrm{Sgn}(1\ 1\ {-1}\ 0)^T = |12\rangle$$

$$[13]^2(0\ 0\ 0\ 1)^T = \mathrm{Sgn}(1\ 1\ {-1}\ 3)^T = |13\rangle \tag{18}$$

(4) four missing bits

$$[13]\ (0\ 0\ 0\ 0)^T = \mathrm{Sgn}\ (0\ 0\ 0\ 0)^T = (-1 -1 -1 -1)^T = |0\rangle$$

$$[13]^2(0\ 0\ 0\ 0)^T = \mathrm{Sgn}\ (-1 -1\ 3\ -1)^T = |2\rangle$$

$$[13]^3(0\ 0\ 0\ 0)^T = \mathrm{Sgn}\ (-3 -3 +3 -3)^T = |2\rangle \tag{19}$$

which converges to a fixed point that is precisely the bipolar complement to $|13\rangle$. In other words, the phase information is lost as an overall minus sign in the last case.

The following sequence of reversing successively from the bipolar bits illustrate tolerance of erroneous bits.

(1) one erroneous bit.

$$[13](-1\ \ 1\ -1\ 1)^T = Sgn(3\ 1\ -1\ 1)^T = |13> \tag{20}$$

(2) two erroneous bits.

$$[13](-1\ -1\ -1\ 1)^T = Sgn(1\ 1\ 1\ -1)^T = |14>$$

$$[13]^2 (-1\ -1\ -1\ 1)^T = Sgn\ (-1\ -1\ -1\ 1)^T = |1>$$

$$[13]^3(-1\ -1\ \ -1\ 1)^T = Sgn\ (1\ 1\ 1\ 1)^T = |15>$$

$$[13]^4(-1\ -1\ \ -1\ 1)^T = Sgn\ (1\ 1\ -3\ 1)^T = |13> \tag{21}$$

(3) three erroneous bits.

$$[13](-1\ -1\ -1\ \ 1)^T = Sgn(-1\ -1\ 1\ -3)^T = |2> \tag{22}$$

which also converges to a fixed point that is also the bipolar complement of $|13>$.

Generalization within a subspace

We consider the ability to recognize a new vector that is different from the stored vec In other words, an AM can recognize its related vectors that has not been memorized before. I recognition, we mean convergence to a different fixed point. In this sense, we say that the AM c generalize its memory to include other fixed points.

In the case of bipolar vectors, if and only if a new vector **x** is orthogonal to the stor vectors, associative recall "converges in a cycle of two" as defined in the following iterations:

$$Sgn([T]\ |x>) = -|x> \tag{23a}$$

$$Sgn(-[T]\ |x>) = +|x> \tag{23b}$$

This necessary and sufficient condition allows us to determine efficiently the orthogonali between a new vector and all the stored vectors.

We shall show that when a new vector $|11>$ is presented to the AM [13], due to t orthogonality between $|13>$ and $|11>$ and traceless property of [13],

$$[13]\ |11> = Sgn(-|11>) = |4>, \text{and}$$

$$[13]^2 |11> = |11> \tag{24}$$

Once the system has acknowledged the second vector $|11>$, it is incorporated into the matrix storage.

$$4\,[\,T_2\,] = [13,11] = [13] + [11]$$
$$= |13><13| + |11><11| - 2\,\delta \qquad (25)$$

If another vector, $|7>$ is presented,

$$[13,11]\ |7> = \mathrm{Sgn}(-2\ |7>) = |8>, \text{ and}$$

$$[13,11]^2|7> = \mathrm{Sgn}(4\ |7>) = |7> \qquad (26)$$

Thus, we enlarge the memory storage to have three memorized states.

$$4[\,T_3\,] = [13,11,7] = [13] + [11] + [7\,] =$$
$$|13><13| + |11><11| + |7><7| - 3\,\delta \qquad (27)$$

This process is continued until the 4-bit orthogonal subspace (p=3) is filled up.

$$4\,[\,T_4\,] = [13] + [11] + [7\,] + [14] \qquad (28)$$

We have demonstrated the ability to include other orthogonal vectors that have not been stored before. This example also shows the important consequence of traceless storage through its contribution to the "generalization by interpolation within the orthogonal subspace".

Given a table of orthogonal vectors, one may argue that computing inner products will also determine orthogonality. However, inner products must be done pairwise among all vectors and become inefficient as the number of vectors gets large. The above method remains efficient for all sizes.

One may furthermore argue that the difficulty is not how to construct orthogonal set, but to select critical bipolar features from gray-scale, imperfect images.

Algorithms for Construct A Critical Feature :

We shall not rely on the auto-AM to select features. One can carry out one's favorite image processing procedure to extract a set of gray-scale feature vectors, $\{|F>\}$. Bipolar feature vectors are preferred in AM because of demonstrated fault-tolerance and the special ability of traceless outer product that allow a quick convergence to a fixed point of cycle two. Given a gray-scale feature vector $|F>$, several procedures for generating a bipolar feature vector are given. The first procedure is "bipolarization", i.e. ,

$$|f> = \mathrm{Sgn}\,(\ |F> - \text{threshold}\,) \qquad (29)$$

The second procedure is to use the Walsh transform. We apply two-dimensional W transform (as orthogonal bipolar vector space{ $|w_i>$ }) to all gray-scale features. We select bipolar feature vector from a specific Walsh base vector that is associated with the maxim coefficient in the Walsh transform.

$$|f> = Sgn(Max_i\ (\Sigma\ |w_i>< w_i|F>) - threshold) \qquad (30)$$

where the orthonormality condition of Walsh base vectors is inserted to relate to the first meth

$$\Sigma\ |w_i>< w_i| = [1] \qquad (31)$$

The third and the fourth procedures are to extract from the arbitrary feature vector | the closest vector $|g>$ from either the bipolar orthogonal feature set {|N> } or the { |F> } using following traceless associative memory storage.

$$|g> = Sgn(\ [\Sigma\Sigma|N>< F|]\ |G> - threshold) \qquad (32)$$

$$|g> = Sgn\ (\ \Sigma\ c_F\ [\,|F> <F\,]|\ G> - threshold) \qquad (33)$$

The linear combination coefficients { c_F } may be determined by the statistics of **sin** **character distortions and variances** (similar to finding the normal modes that diagonalizes covariance matrix and the **Karhunen-Loeve** orthogonal procedure used for outer prod representation of 2-D imagery). Furthermore, the statistics of **character pair distortions**, suc **two scripted zeros,** could be used to determine the coefficients so as to resolve the proble recognizing ***connected character*** and ***broken character*** after segmentation. We will not go i details in this approach, because of its problem-dependent nature.

The mechanism to select critical features is given as follows.

(1) Human being picks a critical feature (pictures) among the set of distorted, handwrit characters, e. g. the extra stroke among ***O, P, Q*** .

(2) Walsh transform the selected feature.

(3) Pick the Walsh function that has the largest transform value.

We choose a feature vector that is closest to the Walsh vector associated with the largest Wa transform coefficient, and the rest follows from the procedure described in eq (24-28). We call s a set of features the critical features.

Lessons to be learned about applying associative memory to pattern recognition:

AM can only do so much. There is no way to judge the correctness of an associative re except by the convergence to a fixed point. One can only assign meaning to those fixed poi whether it is new or old. The proven capabilities of the AM model are (1) missing and errone

bits recovery, and (2) the creation of new orthogonal vectors, as illustrated above. Therefore, to apply AM to pattern recognition, one must apply human interpretations to those capabilities.

Since learning is by trial and error, it is a continuous process. Suppose that a feature vector with many components representing many features (such as leg-feature and fur-feature, etc, for a tiger, coded fully as $|13>$) has been memorized by the traceless outer product. Furthermore, suppose that only certain features are known in a sequence of imperfect input vectors. (I. e., some feature values are missing. e. g. , the first in the sequence is (0, 0, 1, 1)). Then, the AM can fill in the missing bits. After three iterations, one finds $(-1, -1, 1, -1)^T = |2>$. One can then enlarge the traceless outer product memory to include both vectors, [13, 2]. One examines the second input vectors (0, 0, 1, 1). One can verify that the enlarge memory can indeed recall the vector $|2>$, which correspond to, say, a lady, rather than a tiger. The AM "mental" capacity of recognizing other distinct objects when they show up has been demonstrated. Following this line of thought, the different subspace of different size could be assigned for different classes of objects related by a hetero-associative memory of a rectangular matrix. Such a recognition of different classes requires a complete feature set coded in the AM. It can fill all orthogonal subspaces by the "generalization procedure" illustrated in Eq(24-28).

3. ATTENTIVE ASSOCIATIVE MEMORY

Recently, Amari et al has studied the dynamics of such a system, which we will give a simple theorem. We summarize our model equations as follows:

$$< n| m > \equiv B\, \delta_{n,m} \tag{34}$$

$$[T]\, |n> = \lambda_n\, |n> \tag{35}$$

The simple model of attentive associative memory $[T]$ is a linear combination of outer products based on the set of orthogonal feature vectors, $\{ |n> , n = 1, \dots M\}$, and a cue of initial state $|Q>$ that determines the set of attention parameters $\{ a_n \}$ as follows:

$$a_n = <n\,|\,n> - <n\,|\,Q> \tag{36}$$

$$B\,[T]_{ij} = \Sigma^M{}_{n=1}\; a_n\, |\, n_i ><n_j\,| \;-\; A\, [\delta_{i,j}] \tag{37}$$

that is traceless, $Tr\, \delta_{i,j} = Tr\, |\, n_i ><n_j\, | = B$, giving

$$A \equiv \Sigma^M{}_{n=1}\; a_n \tag{38}$$

and

$$\lambda_n = a_n - (A/B) \tag{39}$$

The attentive memory capacity A and eigenvalue λ_n are reduced to Hopfield's memory capacity M and a degenerate eigenvalue λ, in case of a uniform attention(i.e. $a_n = 1$),

$$\lambda \equiv 1 - r \tag{40}$$

where Amari's pattern ratio $r \equiv (M/B)$ is defined for M bipolar words (states) of B bits (neurons) each.

The dynamics is assumed to be governed by matrix-vector inner product

$$Q(t+1) \equiv Sgn([T]\, Q(t)) \tag{41}$$

where a point nonlinear ity function is defined as $Sgn(x) = +1$ if $x > 0$, and -1 if $x < 0$. succesive associative recall gives the iteration, indexed by $t = 0, 1, 2, \ldots$, such that $Q(t) = Q$ when 0.The eigenvalue spectrum, not the distance alone, is a proper macroscopic parameter to explain transient dynamical behaviors of the recalling process. In particular, the direction cosine

$$S_m(t)) \equiv \langle m | Q(t) \rangle / \langle m | m \rangle \tag{42}$$

has been derived and the logarithmic derivative is given by

$$(d/dt) \log(1 - S_m(t)) < \log(\lambda_m / 2) < 0 \tag{43}$$

Convergence to a specific m-th state is guaranteed if m-th eigenvalue (λ_m) is bounded $2 > \lambda_m >$

Theorem 1 about the lower bound says that paying attention (i.e. non-uniform a_n always increases the memory capacity A) $\Sigma^M{}_{n=1}\, a_n > M$ with a faster convergence proportional to the eigenvalue $\lambda_m > \lambda \equiv 1 - r$

We conjecture that the statistical neurodynamics of associative memory may have similar beha to the deterministic dynamics of attentive associative memory with a non-white eigenv spectrum due to random initial conditions that change with respect to the initial guess vector | >, t =0. The difference vector between $|Q(t)\rangle$ from $|m\rangle$ has an inner product norm defined as

$$2\, D_m(t) \equiv \langle m | m \rangle - \langle m | Q(t) \rangle \tag{44}$$

If we assume that paying attention to the initial small guess error $2\, D_m(0)$ amounts to choo nonuniform and biased storage

$$a_m = 2\, D_m(0) \geq 1 \tag{45}$$

and all other coefficients to be identical to 1

$$a_n = 1 \quad , n \neq m. \tag{46}$$

By definition

$$A = M + 2 D_m(0) - 1. \tag{47}$$

Theorem 2 about the upper bound of λ_m assumes that if a small difference vector betweer the input $|Q>$ and the specific state $|m>$, is used as the attention parameter a_m, Eq(31a), then th critical relationship between the Amari's pattern ratio r and the initial error is analytically founc for successful recalls.

$$2 D_m(0) < 2 + (M + 1)/ (B - 1) \tag{48}$$

The maximum permissible Hamming distance D_H, from the desired **m**-th state to b reached after iterative recalls, is given by the formula

$$D_H \leq (B/2) - 1 -[(M- 1)/ 2 (B + 1)] ((B/2) - 1 - (r/2) \tag{49}$$

4. Conclusion

Associative memory (AM) works like a match filter , but does so efficiently. It should not be applied to image domain directly. Rather, it should be applied to feature domain so that a relatively small AM can do useful tasks at the present technology.

We shall not rely on the auto-AM to select features. Instead, features should be selected using human judgement. However, auto-AM will help us find critical features and hetero-associative memory can perform feature extraction efficiently.

There exists a large body of knowledge pertaining to features selection and extraction and pattern classification for traditional optical character recognition in the literature. This body of knowledge should be tapped and coupled with associative memory. One should not rule out the use of traditional classification techniques (such as syntactical) as extraction of high-level features which then become part of the input feature vector to an AM.

Classical pattern recognition has been demonstrated with a relatively greater success in machine-printed character recognition compared to handprinted character recognition. Difficulty may be rooted in the lack of generalization and abstraction due to machine's limited one-dimensional knowledge representation. In principle, AM should be able to complement traditional OCR with 2-D knowledge representation. Various degrees of abstraction can be achieved through a multi-layer, two-dimensional AM architecture. Note that the present technology has evolved to the point where 2-D memory (chip or optical disk) is not more expensive than 1-D memory storage with logic unit tree content addressable memory processor.

In conclusion, we can combine traditional wisdom in traditional OCR with simple implementable in present technology to form a human-intelligence-endowed neu network.

Character segmentation is an important step in character recognition. Fukushima h developed neural network model (selective attention) for character segmentation in h Neocognitron [Fukushima (1987)]. The attentive associative memory model implement opto-electronically by Athale, Szu & Friedlander (1986) can be augmented by a priori probabili compiled by a character-pair correlation function of connected characters. This is an interesti area for more research.

Inputs to associative memory are linear vectors whereas inputs to OCR are rectangul arrays. Can associative memory replicate the concept of (2-D) neighborhood? The tw dimensional transform that preserves the neighborhood relationship should be used for ima pre-processing before applying AM to the pattern. For example, 2-D Walsh transform can gi a 1-D base Walsh vector (associated with the largest coefficient) as input feature vector to t AM.

Can AM perform syntactical parsing [Ali and Pavlidis (1977)] or rule-based structu analysis [D'Amato (1982)]? Any traditional classification technique can be used to extract hig level features for AM.

How can AM extract position and rotation invariant features? [cf. Szu (1986), Messner a Szu (1987)].

One difficulty in applying backpropagation network has been network size-scali problem. One way to circumvent it has been to extract a small number of features as input. [Burr (1987), Gullichsen and Chang (1987)]. Recent advances by Ballard in 1987 permit parti connectivity between two successive layers which avoids combinatorial explosions oft encountered when the input layer is directly connected to image pixels. Thus, spatial patte relationship can be efficiently preserved in such a network while coarse-graining betwe successive layers can desensitize pattern variation in input images.

An AI extension of the simple AM model is attentive associative memory, (AAM), th allows us to apply AI to pay a non-uniform attention to each term of outer product storage, i a linear combination of outer products in which the set of combination coefficients determined by AI rule-based system, e.g. the frequency distribution of distorted characte according to user group profiles, e.g. left hand writing versus righthand writing. The efficien of the closed system of rule-based knowledge representation of AI using the tuple storage combined with the flexibility of the non-rule based open system using the matrix knowled representation of NI (coined for either neural, or network, or natural intelligence). Thus, t ability of generalization and abstraction becomes possible for AI, and is demonstrated in combined intelligent system of AI & NI. We can endow a simple neural network architectu based on a small set of neurons with a human-like intelligence by combining the tradition rule-based AI wisdom with non-rule-based learning. This is achievable because OCR requir

better feature vectors obtained from other discipline in the sense of fault tolerance that neural networks built at the present technology can already provide with.

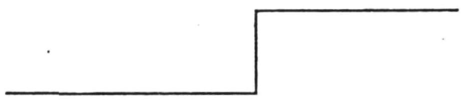
Appendix: Generic Definition of Neural Networks

Associative memory is a special model of neural networks. Examples of associative recalls from partial images and the success of nonlinear signal processing are recorded in the literature [cf. Kohonen (1984)]. An axiomatic definition is outlined as follows.

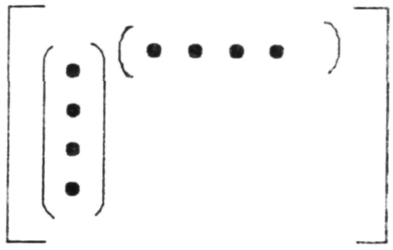
We shall define **three kinds of neurons**: fine-grained, medium-grained and large-grained processor elements (PEs). A fine-grained PE, represented by the lower case word ***neuron***, has no internal memory analogous to neurons in the hippocampus part of the brain that is responsible for fault-tolerant associative recall. A medium-grained PE, ***Neuron,*** has a built-in memory analogous to Neurons in biological sensory and motor control which are responsible for reactions to approaching danger. A large-grained PE, ***NEURON***, has built-in memory, control logic, and communication capabilities equivalent to a computer. NEURONs occur in nature in the form of grandmother cells or pacer/conductor cells.

These three types of neurons and their associated circuits have **four kinds of interactions:** ***(1) exciting, (2) inhibiting, (3) bursting, (4) grading and delaying transmission.*** In general they follow the law of the middle response or the sigmoid function (hyperbolic tangent or logistic functions) to amplify weak signals with a nonlinear quick rising function and suppress strong signals with a nonlinear tapering off saturation function. The generic definition of a Neural Network is a system which is:

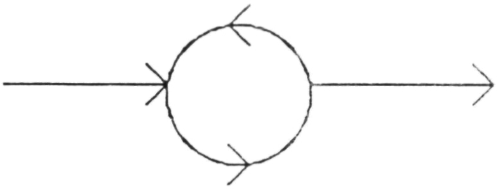
1. ***Non-linear*** ≈ sigmoid function ≈ point non-linearity (hard limiting) shown as follows:



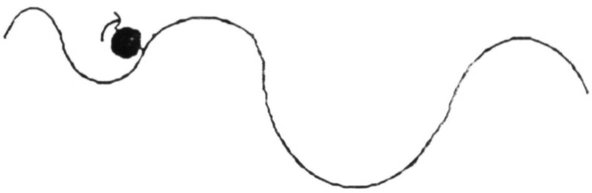
2. ***Non-local*** ≈ weighted outer product ≈ outer product (white spectrum) shown as follows:



3. *Non-stationary* ≈ piecewise time stationary ≈ iterative algorithm shown as follows:



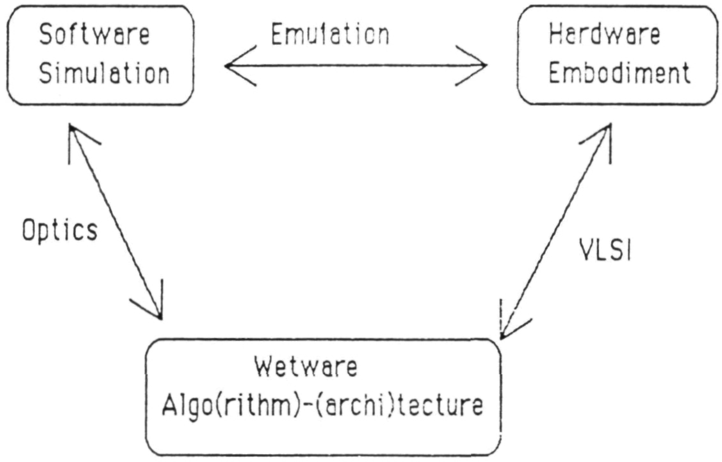
4. *Non-convex* ≈ constrained global optimization ≈ simulated annealing schematica shown as follows:



5. *Other attributes yet to be discovered* .

These successive approximations of the four *non*--principles, indicated by wiggly equal signs in (1-4), makes possible the unveiling of the complex and nonlinear neural (bra behavior. This is possible with the use of powerful computers and more accurate models intelligent functions. The theory is amenable to numerical simulations due to ***piecewi linear, regionally local, temporarily stationary, and locally convex approximations.***

Three decades ago, Rosenblatt and co-workers built the **perceptron** solely based upon t first attribute (nonlinearity) with stochastic implementations. Thus, with hindsight, it was r surprising that Minsky and Papert could show a limited utility and propose useful alternati **artificial intelligence (AI) rule- based systems.** ***AI works in closed systems where rules gove while neural intelligence (NI) works in open systems where rules have yet to be discover*** Various exploitation of these efforts in neural networks are:



The term wet-ware, coined by Carver Mead, is neither software nor hardware, but more like a Hecht-Nielsen's net-ware based on non-programmable but trainable networks. A special version of layered neural networks has been demonstrated with the ability of phonetic interpolation in the Rumelhart, Sejnowski connectionist's networks, such as Net-Talk, Boltzmann and Cauchy Machines, and error back propagation networks.

Acknowledgement

The work has been supported by ONR under IST/SDIO program. Discussions with John Tan and Frank Polkinghorn are gratefully acknowledged.

6. REFERENCES

Ahmed, P. and Suen, C., "*Computer recognition of totally unconstrained handwritten ZIP Codes,*" International Journal of Pattern Recognition and Artificial Intelligence,Vol. 1 (1987), pp.1-15.

Ali, F. and Pavlidis, T., "*Syntactic recognition of handwritten numerals,*" IEEE Transactions on SMC, Vol. 7 (1977), pp.537-541.

Athale, R.A., Szu, H.H., & Friedlander, C.B., "*Optical implementation of associative memory with controlled nonlinearity in the correlation domain,*" Optics Letters, Vol. 11 (1986), pp. 482-484

Burr, D., "*Designing a handwriting reader,*" IEEE Trans. on PAMI, Vol.5 (1983)

Burr, D., "*Experiments with a connectionist text reader,*" Proc. of IEEE International Conference on Neural Networks, 1987, Vol. IV, pp. 717-724.

D'Amato, D., et al, "*High speed pattern recognition system for alphanumeric handprinted characters,*" Proc. of Pattern Recognition and Image Processing, 1982, pp.165-171.

Duda, R. and Hart, P., *Pattern Classification and Scene Analysis* Wiley-Interscience, 1973.

Duerr, B., Haettich, W., Tropf, H. and Winkler, G., "*A combination of statistical and syntacti pattern recognitionapplied to classification of unconstrained handwritten numerals,*" Patte Recognition, Vol. 12 (1980), pp.189-199.

Fukushima, K. and Miyake, S., "*Neocognitron: a new algorithm for pattern recognition tolerant deformations and shifts in position,*" Pattern Recognition, VOL. 15, 1982, pp. 455-469.

Fukushima, K., "*A neural network model for selective attention,*" in Proc. of IEEE Internatio Conference on Neural Networks,1987, Vol. II, pp. 11-18.

Gullichsen, e. and Chang, E., "*Pattern classification by neural network: an experiment system icon recognition,*" Proc. of IEEE International Conference on Neural Networks, 1987, Vol. I pp.725-732.

Hull, J., et al., *Optical Character Recognition Techniques in Mail Sorting: A Review of Algorithm* Technical Report 214, State University of New York at Buffalo, Department of Computer Scien 1984.

Kohonen, T.,"*Self-Organization and Associative Memory,*"Springer-Verlag, 1984.

Stringa, L., "*LCD: a formal language for constraint)free hand-printed character recognition,*" P of International Conference on Pattern Recognition, 1978, pp. 354-358.

Schurmann, J., "*Multifont word recognition system with application to postal address readin* Proc. of Int. Conf. on Pattern Recognition, 1976, pp. 658-662.

Schurmann, J., "*Reading machines*", Proc. of Int. Conf. on Pattern Recognition, 1982, pp.1031-104

Shridhar, M. and Badreldin, A., "*Recognition of isolated and simply connected handwrit numerals,*" Pattern Recognition, Vol.19 (1986), pp.1-12.

Suen, C. Y., Berthod, M. and Mori, S., "*Automatic recognition of handprinted characters - the st of the art*", Proceedings of the IEEE, vol. 68 (1980), pp. 469-487.

Szu, H. H., Caulfield, H.J., "*Optical Expert Systems,*" Applied Optics, Vol. 26, pp. 1943-1947, 1987

Szu, Harold H., "*Three layers of vector outer product neuural networks for optical patte recognition.*", In H.Szu (Ed.) *Optical and Hybrid computing* (1986) (pp. 312-330), Bellingham, W Society of Photo-Optical Instrumentation Engineers.

Szu, H. H. & Messner, R. A. ,"*Adaptive Invariant Novelty Filters,*" Proceedings of IEEE, Vol. (1986), pp. 518-519

Szu, Harold H., "*Globally connected network models for computing using fine-grained processing elements*," In C. P. Wang (ed.) Proceedings of International Conference on LASERS '85, pp. 92-97, Society for Optical & Quantum Electronics, P.O. Box 245, McLean VA 22101

Szu, Harold H., and Tan, John,"Can associative memory recognize charactors ?",
Third Advanced Technology Conference, U.S. Postal Service, Washington D.C. May 3-5, 1988

Tou, J., and Gonzalez, R., *Pattern Recognition Principles,* Addison-Wesley, 1974.

Winston, P. H. , "*Artificial Intelligence,*" Reading , Mass. Addison-Wesley 1984, 2-ed.

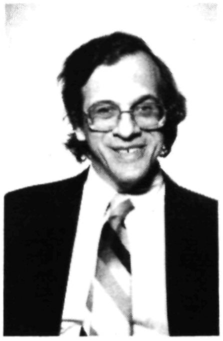
NOTES

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

59

59-61
ABS. ONLY
15389

N91-71359



Daniel S. Levine

University of Texas
Arlington, Texas

Dr. Levine received his Ph.D. in applied mathematics at MIT in 1975, under Professor Stephen Grossberg. He has been an associate professor of mathematics at the University of Texas at Arlington since the fall of 1983, and his research articles have been in the theory, modeling, and simulation of biological (neural or ecological) processes using dynamical systems of differential equations. In neural networks, Dr. Levine has written an 86-page review article (Mathematical Biosciences, Vol. 66, 1983) and is currently writing a graduate textbook (Introduction to Neural and Cognitive Modeling, Erlbaum, to appear in 1989). He is an editor of Neural Networks and has been on the organizing committees of both 1987 and 1988 International Neural Network Society meetings. His recent work has centered on interactions between cognitive and motivational variables in neural networks. Dr. Levine has also been a commissioned officer at the National Institute of Health; a postdoctoral trainee in physiology at UCLA; and an assistant professor of mathematics at Rutgers University, the University of Pittsburgh, and the University of Houston.

NEURAL MODELING OF SELECTIVE ATTENTION

Abstract

A neural network is presented in which there are modifiable, bidirectional connections between nodes representing sensory events and other nodes representing reinforcement sources. There is also competition between sensory nodes. Through these competitive and associative mechanisms, the presenter, together with Stephen Grossberg, has simulated some data on attentionally modulated Pavlovian conditioning. In particular, if two stimuli are presented simultaneously, and one of them has already been associated with a primary reinforcer (such as electric shock or food), selective attention occurs which inhibits the other stimulus from forming new associations. Context changes can profoundly alter the dynamics of selective attention. For example, if one stimulus has been paired with a reinforcer and that stimulus combined with another is paired with a greater or lesser amount of that reinforcer, the second stimulus is no longer blocked. Also, selective attention based on positive or negative reinforcement can compete with selective attention based on other criteria. Nonmotivational criteria are enhanced by frontal lobe damage, which weakens the sensory-reinforcement linkage. For example, a frontally lesioned monkey can prefer a novel object to one that has previously been rewarded. Also, a human frontal lobe patient can persevere in a habit that was once, but is no longer, rewarding.

NOTES

S10-62
153 10

N91-71360

P-14

James M. Bower, Ph.D.

California Institute of Technology
Pasadena, California

Dr. Bower is an assistant professor of biology at CalTech. He attended Antioch College; Montana State University (B.S. in zoology); University of Wisconsin-Madison (Ph.D. in neurophysiology); New York University Medical Center; and The Marine Biological Laboratory, Woods Hole, MA (postdoctoral fellow). Honors include the Jerzy E. Rose Award for original and significant research in the neural sciences, 1981; and the National Research Service Award from NINCDS, 1981 to 1983. Research interests focus on a computational approach to studying cerebellar and olfactory cortices using neuroanatomical, neurophysiological, and computer modeling techniques. Dr. Bower is also center forward on the Caltech ice hockey team; lead singer for "Ramon and the K Halls"; and founder and codirector of Project Seed, a new approach to elementary school science instruction.

APPLIED AND REAL NEURAL NETWORKS: A COORDINATED AND INTERDEPENDENT INVESTIGATION OF BOTH

Abstract

One and a half years ago, Caltech organized a new graduate program in Computation and Neural Systems (CNS). This program involves 15 faculty members with interests as diverse as statistical physics, concurrent computing, analog VLSI, signal processing, optical computing, machine vision robotics, and the neurobiological study of numerous real neural systems. The Bower laboratory is an integral part of the CNS program with our primary interest being the coordinated study of information processing in real neural networks. The principal approach taken is one that views these networks as systems of complex processing elements having functions that are intimately related to their specific distributed architectures. Within the lab, our multidisciplinary approach includes standard anatomical and physiological investigations linked to computer modeling techniques. In addition, we are developing new experimental techniques which directly address computational issues in real neural structures. For example, we are using modern silicon manufacturing technology to make multisite brain recording electrodes which capture the activity of multiple, functionally related neurons. We have also constructed a general-purpose neural network simulator with interactive graphics (CAD/CAM for neural networks) that runs on concurrent computers. Finally, we have been exploring the use of applied neural networks in recognizing and categorizing recorded signals from real neurons. Each of these efforts is described. This work is sponsored by the Whitaker Foundation, the Joseph Drown Foundation, and Lockheed Corporation.

To appear in:

Neural Information Processing Systems
D. Anderson (ed.)
AIP Press

A Computer Simulation of Olfactory Cortex With Functional Implications for Storage and Retrieval of Olfactory Information

Matthew A. Wilson and James M. Bower
Computation and Neural Systems Program
Division of Biology, California Institute of Technology, Pasadena, CA 91125

ABSTRACT

Based on anatomical and physiological data, we have developed a computer simulation of piriform (olfactory) cortex which is capable of reproducing spatial and temporal patterns of actual cortical activity under a variety of conditions. Using a simple Hebb-type learning rule in conjunction with the cortical dynamics which emerge from the anatomical and physiological organization of the model, the simulations are capable of establishing cortical representations for different input patterns. The basis of these representations lies in the interaction of sparsely distributed, highly divergent/convergent interconnections between modeled neurons. We have shown that different representations can be stored with minimal interference, and that following learning these representations are resistant to input degradation, allowing reconstruction of a representation following only a partial presentation of an original training stimulus. Further, we have demonstrated that the degree of overlap of cortical representations for different stimuli can also be modulated. For instance similar input patterns can be induced to generate distinct cortical representations (discrimination), while dissimilar inputs can be induced to generate overlapping representations (accommodation). Both features are presumably important in classifying olfactory stimuli.

INTRODUCTION

Piriform cortex is a primary olfactory cerebral cortical structure which receives second order input from the olfactory receptors via the olfactory bulb (Fig. 1). It is believed to play a significant role in the classification and storage of olfactory information[1,2,3]. For several years we have been using computer simulations as a tool for studying information processing within this cortex[4,5]. While we are ultimately interested in higher order functional questions, our first modeling objective was to construct a computer simulation which contained sufficient neurobiological detail to reproduce experimentally obtained cortical activity patterns. We believe this first step is crucial both to establish correspondences between the model and the cortex, and to assure that the model is capable of generating output that can be compared to data from actual physiological experiments. In the current case, having demonstrated that the behavior of the simulation at least approximates that of the actual cortex[4] (Fig. 3), we are now using the model to explore the types of processing which could be carried out by this cortical structure. In particular, in this paper we will describe the ability of the simulated cortex to store and recall cortical activity patterns generated by stimulus various conditions. We believe this approach can be used to provide experimentally testable hypotheses concerning the functional organization of this cortex which would have been difficult to deduce solely from neurophysiological or neuroanatomical data.

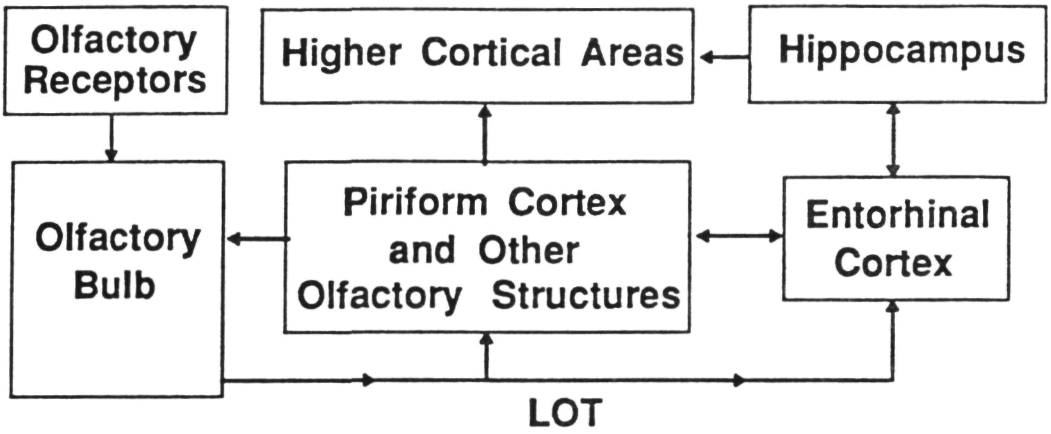


Fig. 1. Simplified block diagram of the olfactory system and closely related structures.

MODEL DESCRIPTION

This model is largely instructed by the neurobiology of piriform cortex[3]. Axonal conduction velocities, time delays, and the general properties of neuronal integration and the major intrinsic neuronal connections approximate those currently described in the actual cortex. However, the simulation reduces both the number and complexity of the simulated neurons (see below). As additional information concerning the these or other important features of the cortex is obtained it will be incorporated in the model. Bracketed numbers in the text refer to the relevent mathematical expressions found in the appendix.

Neurons. The model contains three distinct populations of intrinsic cortical neurons, and a fourth set of cells which simulate cortical input from the olfactory bulb (Fig. 2). The intrinsic neurons consist of an excitatory population of pyramidal neurons (which are the principle neuronal type in this cortex), and two populations of inhibitory interneurons. In these simulations each population is modeled as 100 neurons arranged in a 10x10 array (the actual piriform cortex of the rat contains on the order of 10^6 neurons). The output of each modeled cell type consists of an all-or-none action potential which is generated when the membrane potential of the cell crosses a threshold [2.3]. This output reaches other neurons after a delay which is a function of the velocity of the fiber which connects them and the cortical distance from the originating neuron to each target neuron [2.0, 2.4]. When an action potential arrives at a destination cell it triggers a conductance change in a particular ionic channel type in that cell which has a characteristic time course, amplitude, and waveform [2.0, 2.1]. The effect of this conductance change on the transmembrane potential is to drive it towards the equilibrium potential of that channel. Na^+, Cl^-, and K^+ channels are included in the model. These channels are differentially activated by activity in synapses associated with different cell types (see below).

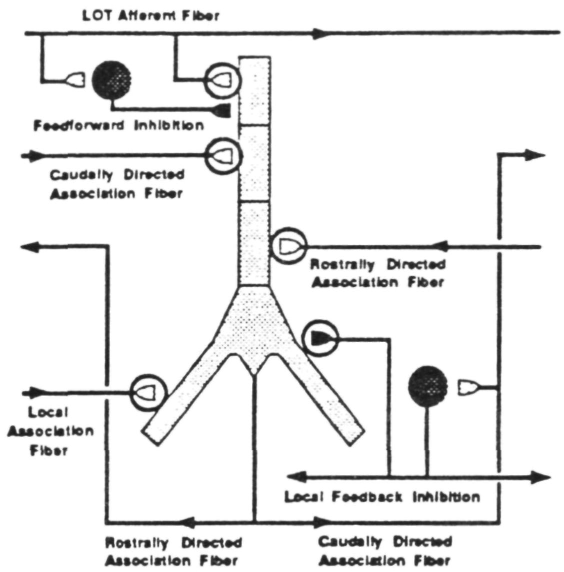


Fig. 2. Schematic diagram of piriform cortex showing an excitatory pyramidal cell and two inhibitory interneurons with their local interactions. Circles indicate sites of synaptic modifiability.

Connection Patterns. In the olfactory system, olfactory receptors project to the olfactory bulb which, in turn, projects directly to the piriform cortex and other olfactory structures (Fig. 1). The input to the piriform cortex from the olfactory bulb is delivered via a fiber bundle known as the lateral olfactory tract (LOT). This fiber tract appears to make sparse, non-topographic, excitatory connections with pyramidal and feedforward inhibitory neurons across the extent of the cortex[3,6]. In the model this input is simulated as 100 independent cells each of which make random connections ($p=0.05$) with pyramidal and feedforward inhibitory neurons (Fig. 1 and 2).

In addition to the input connections from the olfactory bulb, there is also an extensive set of connections between the neurons intrinsic to the cortex (Fig. 2). For example, the association fiber system arises from pyramidal cells and makes sparse, distributed excitatory connections with other pyramidal cells all across the cortex[7,8,9] . In the model these connections are randomly distributed with 0.05 probability. In the model and in the actual cortex, pyramidal cells also make excitatory connections with nearby feedforward and feedback inhibitory cells. These interneurons, in turn, make reciprocal inhibitory connections with the group of nearby pyramidal cells. The primary effect of the feedback inhibitory neurons is to inhibit pyramidal cell firing through a Cl^- mediated current shunting mechanism[10,11,12]. Feedforward interneurons inhibit pyramidal cells via a long latency, long duration, K^+ mediated hyperpolarizing potential[12,13]. Pyramidal cell axons also constitute the primary output of both the model and the actual piriform cortex[7,14].

Synaptic Properties and Modification Rules. In the model, each synaptic connection has an associated weight which determines the peak amplitude of the conductance change induced in the postsynaptic cell following presynaptic activity [2.0]. To study learning in the model, synaptic weights associated with some of the fiber systems are modifiable in an activity-dependent fashion (Fig. 2). The basic modification rule in each case is Hebb-like; i.e. change in synaptic strength is proportional to presynaptic activity multiplied by the offset of the postsynaptic membrane potential from a baseline potential. This baseline potential is set slightly more positive than the Cl^- equilibrium potential associated with the shunting feedback inhibition. This means that synapses activated while a destination cell is in a depolarized or excited state are strengthened, while those activated during a period of inhibition are weakened. In the model, synapses which follow this rule include the association fiber connections between excitatory pyramidal neurons as well as the connections between inhibitory neurons and pyramidal neurons. Whether these synapses are modifiable in this way in the actual cortex is a subject of active research in our lab. However, the model does mimic the actual synaptic properties associated with the input pathway (LOT) which we have shown to undergo a transient increase in synaptic strength following activation which is independent of postsynaptic potential[15]. This increase is not permanent and the synaptic strength subsequently returns to its baseline value.

Generation of Physiological Responses. Neurons in the model are represented as first-order "leaky" integrators with multiple, time-varying inputs [1.0]. During simulation runs, membrane potentials and currents as well as the time of occurence of action potentials are stored for comparison with actual data. An explicit compartmental model (5 compartments) of the pyramidal cells is used to generate the spatial current distributions used for calculation of field potentials (evoked potentials, EEGs) [3.0, 4.0].

Stimulus Characteristics. To compare the responses of the model to those of the actual cortex, we mimicked actual experimental stimulation protocols in the simulated cortex and contrasted the resulting intracellular and extracellular records. For example, shock stimuli applied to the LOT are often used to elicit characteristic cortical evoked potentials *in vivo*[16,17,18]. In the model we simulated this stimulus paradigm by simultaneously activating all 100 input fibers. Another measure of cortical activity used most successfully by Freeman and colleagues involves recording EEG activity from piriform cortex in behaving animals[19,20]. These odor-like responses were generated in the model through steady, random stimulation of the input fibers.

To study learning in the model, once physiological measures were established, it was required that we use more refined stimulation procedures. In the absence of any specific information about actual input activity patterns along the LOT, we constructed each stimulus out of a randomly selected set of 10 out of the 100 input

fibers. Each stimulus episode consisted of a burst of activity in this subset of fibers with a duration of 10 msec at 25 msec intervals to simulate the 40 Hz periodicity of the actual olfactory bulb input. This pattern of activity was repeated in trials of 200 msec duration which roughly corresponds to the theta rhythm periodicity of bulbar activity and respiration[21,22]. Each trial was then presented 5 times for a total exposure time of 1 second (cortical time). During this period the Hebb-type learning rule could be used to modify the connection weights in an activity-dependent fashion.

Output Measure for Learning. Given that the sole output of the cortex is in the form of action potentials generated by the pyramidal cells, the output measure of the model was taken to be the vector of spike frequency for all pyramidal neurons over a 200 msec trial, with each element of the vector corresponding to the firing frequency of a single pyramidal cell. Figures 5 through 8 show the 10 by 10 array of pyramidal cells. The size of the box placed at each cell position represents the magnitude of the spike frequency for that cell. To evaluate learning effects, overlap comparisons between response pairs were made by taking the normalized dot product of their response vectors and expressing that value as a percent overlap (Fig. 4).

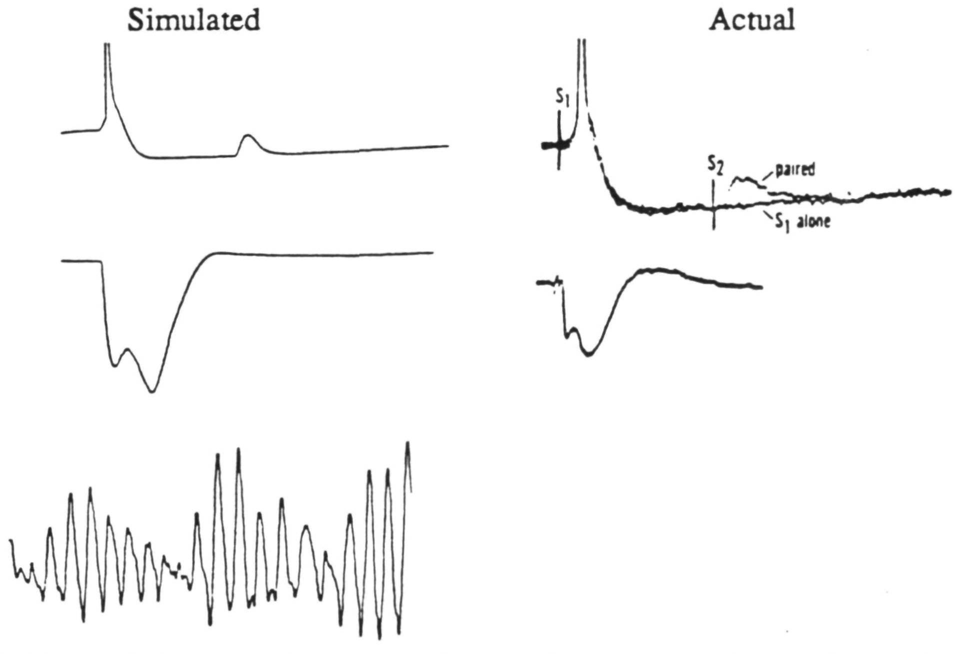


Fig. 3. Simulated physiological responses of the model compared with actual cortical responses. Upper: Simulated intracellular response of a single cell to paired stimulation of the input system (LOT) (left) compared with actual response (right) (Haberly & Bower,'84). Middle: Simulated extracellular response recorded at the cortical surface to stimulation of the LOT (left), compared with actual response (right) (Haberly,'73b). Lower: Stimulated EEG response recorted at the cortical surface to odor-like input (left), for actual EEG see Freeman 1978.

Computational Requirements. All simulations were carried out on a Sun Microsystems 3/260 model microcomputer equipped with 8 Mbytes of memory and a floating point accelerator. Average time for a 200 msec simulation was 3 cpu minutes.

RESULTS

Physiological Responses

As described above, our initial modeling objective was to accurately simulate a wide range of activity patterns recorded, by ourselves and others, in piriform cortex using various physiological procedures. Comparisons between actual and simulated records for several types of response are shown in figure 3. In general, the model replicated known physiological responses quite well (Wilson et al in preparation describes, in detail, the analysis of the physiological results). For example in response to shock stimulation of the input pathway (LOT), the model reproduces the principle characteristics of both the intracellular and location-dependent extracellular waveforms recorded in the actual cortex[9,17,18] (Fig. 3).

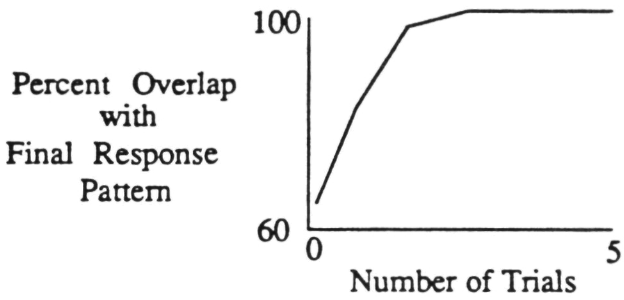


Fig. 4. Convergence of the cortical response during training with a single stimulus with synaptic modification.

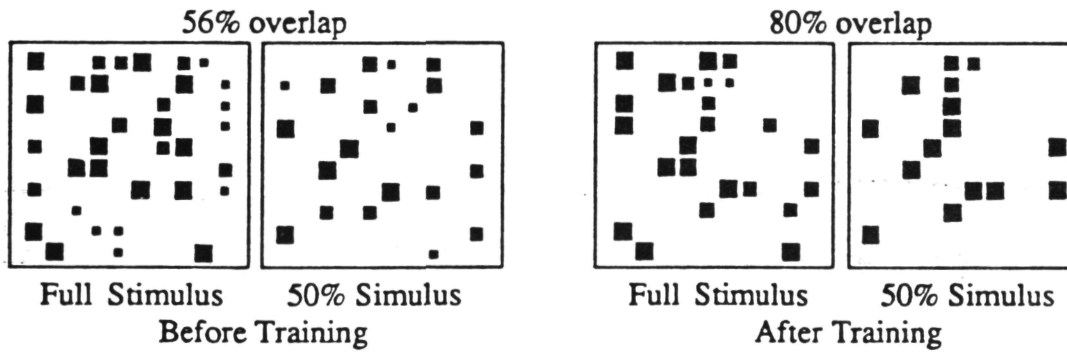


Fig. 5. Reconstruction of cortical response patterns with partially degraded stimuli. Left: Response, before training, to the full stimulus (left) and to the same stimulus with 50% of the input fibers inactivated (right). There is a 44% degradation in the response. Right: Response after training, to the full stimulus (left), and to the same stimulus with 50% of the input fibers inactivated (right). As a result of training, the degradation is now only 20%.

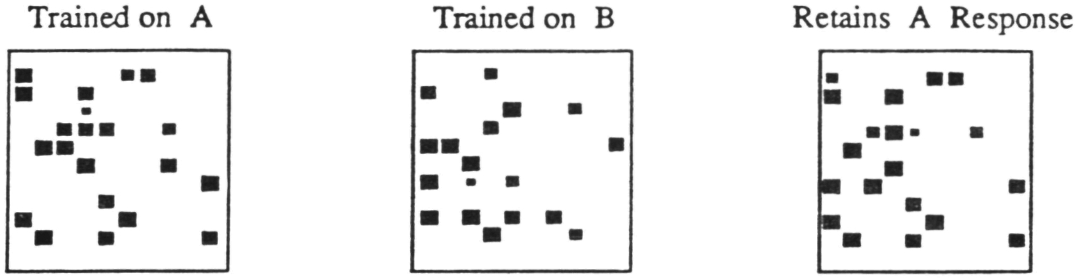


Fig. 6. Storage of multiple patterns. Left: Response to stimulus A after training. Middle: Response to stimulus B after training on A followed by training on B. Right: Response to stimulus A after training on A followed by training on B. When compared with the original response (left) there is an 85% congruence.

Further, in response to odor-like stimulation the model exhibits 40 Hz oscillations which are characteristic of the EEG activity in olfactory cortex in awake, behaving animals[19]. Although beyond the scope of the present paper, the simulation also duplicates epileptiform[9] and damped oscillatory[16] type activity seen in the cortex under special stimulus or pharmacological conditions[4].

Learning

Having simulated characteristic physiological responses, we wished to explore the capabilities of the model to store and recall information. Learning in this case is defined as the development of a consistent representation in the activity of the cortex for a particular input pattern with repeated stimulation and synaptic modification. Figure 4 shows how the network converges, with training, on a representation for a stimulus. Having demonstrated that, we studied three properties of learned responses - the reconstruction of trained cortical response patterns with partially degraded stimuli, the simultaneous storage of separate stimulus response patterns, and the modulation of cortical response patterns independent of relative stimulus characteristics.

Reconstruction of Learned Cortical Response Patterns with Partially Degraded Stimuli. We were interested in knowing what effect training would have on the sensitivity of cortical responses to fluctuations in the input signal. First we presented the model with a random stimulus A for one trial (without synaptic modification). On the next trial the model was presented with a degraded version of A in which half of the original 10 input fibers were inactivated. Comparison of the responses to these two stimuli in the naive cortex showed a 44% variation. Next, the model was trained on the full stimulus A for 1 second (with synaptic modification). Again, half of the input was removed and the model was presented with the degraded stimulus for 1 trial (without synaptic modification). In this case the dif-

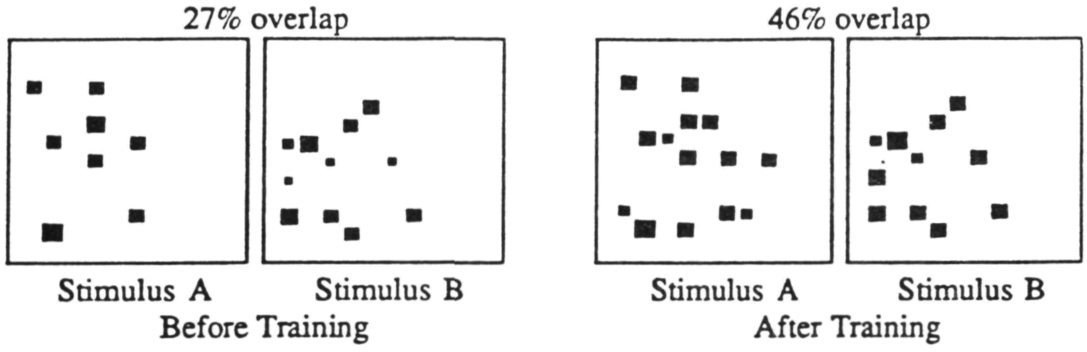


Fig. 7. Results of merging cortical response patterns for dissimilar stimuli. Left: Response to stimulus A and stimulus B before training. Stimuli A and B do not activate any input fibers in common but still have a 27% overlap in cortical response patterns. Right: Response to stimulus A and stimulus B after training in the presence of a common modulatory input E1. The overlap in cortical response patterns is now 46%.

ference between cortical responses was only 20% (Fig. 5) showing that training increased the robustness of the response to degradation of the stimulus.

Storage of Two Patterns. The model was first trained on a random stimulus A for 1 second. The response vector for this case was saved. Then, continuing with the weights obtained during this training, the model was trained on a new non-overlapping (i.e. different input fibers activated) stimulus B. Both stimulus A and stimulus B alone activated roughly 25% of the cortical pyramidal neurons with 25% overlap between the two responses. Following the second training period we assessed the amount of interference in recalling A introduced by training with B by presenting stimulus A again for a single trial (without synaptic modification). The variation between the response to A following additional training with B and the initially saved reponse to A alone was less than 15% (Fig. 6) demonstrating that learning B did not substantially interfere with the ability to recall A.

Modulation of Cortical Response Patterns. It has been previously demonstrated that the stimulus evoked response of olfactory cortex can be modulated by factors not directly tied to stimulus qualities, such as the behavioral state of the animal[1,20,23]. Accordingly we were interested in knowing whether the representations stored in the model could be modulated by the influence of such a "state" input.

One potential role of a "state" input might be to merge the cortical response patterns for dissimilar stimuli; an effect we refer to as accomodation. To test this in the model, we presented it with a random input stimulus A for 1 trial. It was then presented with a random input stimulus B (non-overlapping input fibers). The amount of overlap in the cortical responses for these untrained cases was 27%. Next, the model was trained for 1 second on stimulus A in the presence of an additional random "state" stimulus E1 (activity in a set of 10 input fibers distinct

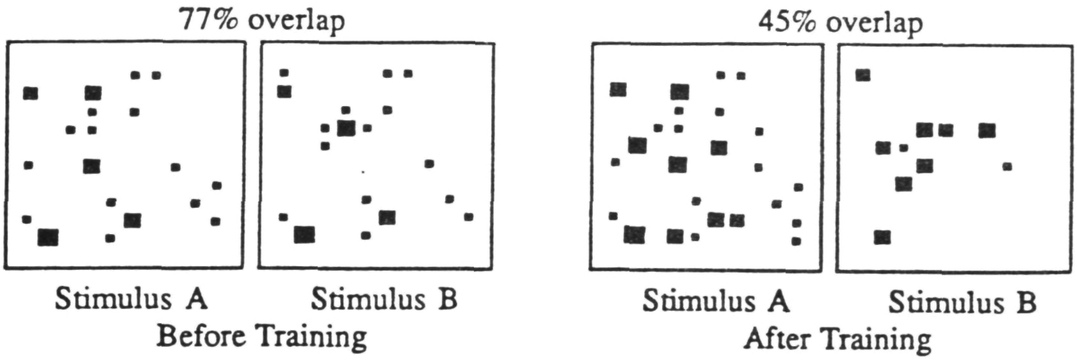


Fig. 8. Results of differentiating cortical response patterns for similar stimuli. Left: Response to stimulus A and stimulus B before training. Stimuli A and B activate 75% of their input fibers in common and have a 77% overlap in cortical response patterns. Right: Response to stimulus A and stimulus B after training A in the presence of modulatory input E1 and training B with a different modulatory input E2. The overlap in cortical response patterns is now 45%.

from both A and B). The model was then trained on stimulus B in the presence of the same "state" stimulus E1. After training, the model was presented with stimulus A alone for 1 trial and stimulus B alone for 1 trial. Results showed that now, even without the coincident E1 input, the amount of overlap between A and B responses was found to have increased to 46% (Fig 7). The role of E1 in this case was to provide a common stimulus component during learning which reinforced shared components of the responses to input stimuli A and B.

To test the ability of a state stimulus to induce differentiation of cortical response patterns for similar stimuli, we presented the model with a random input stimulus A for 1 trial, followed by 1 trial of a random input stimulus B (75% of the input fibers overlapping). The amount of overlap in the cortical responses for these untrained cases was 77%. Next, the model was trained for a period of 1 second on stimulus A in the presence of an additional random "state" stimulus E1 (a set of 10 input fibers not overlapping either A or B). It was then trained on input stimulus B in the presence of a different random "state" stimulus E2 (10 input fibers not overlapping either A, B, or E1) After this training the model was presented with stimulus A alone for 1 trial and stimulus B alone for 1 trial. The amount of overlap was found to have decreased to 45% (Fig 8). In this situation E1 and E2 provided a differential signal during learning which reinforced distinct components of the responses to input stimuli A and B.

DISCUSSION

Physiological Responses. Detailed discussion of the mechanisms underlying the simulated patterns of physiological activity in the cortex is beyond the scope of the current paper. However, the model has been of value in suggesting roles for

specific features of the cortex in generating physiologically recorded activity. For example, while actual input to the cortex from the olfactory bulb is modulated into 40 Hz bursts[24], continuous stimulation of the model allowed us to demonstrate the model's capability for intrinsic periodic activity independent of the complementary pattern of stimulation from the olfactory bulb. While a similar ability has also been demonstrated by models of Freeman[25], by studying this oscillating property in the model we were able to associate these oscillatory characteristics with specific interactions of local and distant network properties (e.g. inhibitory and excitatory time constants and trans-cortical axonal conduction velocities). This result suggests underlying mechanisms for these oscillatory patterns which may be somewhat different than those previously proposed.

Learning. The main subject of this paper is the examination of the learning capabilities of the cortical model. In this model, the apparently sparse, highly distributed pattern of connectivity characteristic of piriform cortex is fundamental to the way in which the model learns. Essentially, the highly distributed pattern of connections allows the model to develop stimulus-specific cortical response patterns by extracting correlations from randomly distributed input and association fiber activity. These correlations are, in effect, stored in the synaptic weights of the association fiber and local inhibitory connections.

The model has also demonstrated robustness of a learned cortical response against degradation of the input signal. A key to this property is the action of sparsely distributed association fibers which provide reinforcment for previously established patterns of cortical activity. This property arises from the modification of synaptic weights due to correlations in activity between intra-cortical association fibers. As a result of this modification the activity of a subset of pyramidal neurons driven by a degraded input drives the remaining neurons in the response.

In general, in the model, similar stimuli will map onto similar cortical responses and dissimilar stimuli will map onto dissimilar cortical responses. However, a presumably important function of the cortex is not simply to store sensory information, but to represent incoming stimuli as a function of the absolute stimulus qualities and the context in which the stimulus occurs. The fact that many of the structures that piriform cortex projects to (and receives projections from) may be involved in multimodal "state" generation[14] is circumstantial evidence that such modulation may occur. We have demonstrated in the model that such a modulatory input can modify the representations generated by pairs of stimuli so as to push the representations of like stimuli apart and pull the representations of dissimilar stimuli together. It should be pointed out that this modulatory input was not an "instructive" signal which explicitly directed the course of the representation, but rather a "state" signal which did not require *a priori* knowledge of the representational structure. In the model, this modulatory phenomenon is a simple consequence of the degree of overlap in the combined (odor stimulus + modulator) stimulus. Both cases approached approximately 50% overlap in cortical responses reflecting the approximately 50% overlap in the combined stimuli for both cases.

Of interest was the use of the model's reconstructive capabilities to maintain the modulated response to each input stimulus even in the absence of the modulatory input.

CAVEATS AND CONCLUSIONS

Our approach to studying this system involves using computer simulation to investigate mechanisms of information processing which could be implemented given what is known about biological constraints. The significance of results presented here lies primarily in the finding that the structure of the model and the parameter settings which were appropriate for the reproduction of physiological responses were also appropriate for the proper convergence of a simple, biologically plausible learning rule under various conditions. Of course, the model we have developed is only an approximation to the actual cortex limited by our knowledge of its organization and the computing power available. For example, the actual piriform cortex of the rat contains on the order of 10^6 cells (compared with 10^2 in the simulations) with a sparsity of connection on the order of p=0.001 (compared with p=0.05 in the simulations). Our continuing research effort will include explorations of the scaling properties of the network.

Other assumptions made in the context of the current model include the assumption that the representation of information in piriform cortex is in the form of spatial distributions of rate-coded outputs. Information contained in the spatio-temporal patterns of activity was not analyzed, although preliminary observation suggests that this may be of significance. In fact, the dynamics of the model itself suggest that temporally encoded information in the input at various time scales may be resolvable by the cortex. Additionally, the output of the cortex was assumed to have spatial uniformity, i.e. no differential weighting of information was made on the basis of spatial location in the cortex. But again, observation of the dynamics of the model, as well as the details of known anatomical distribution patterns for axonal connections, indicate that this is a major oversimplification. Preliminary evidence from the model would indicate that some form of hierarchical structuring of information along rostral/caudal lines may occur. For example it may be that cells found in progressively more rostral locations would have increasingly non-specific odor responses.

Further investigations of learning within the model will explore each of these issues more fully, with attempts to correlate simulated findings with actual recordings from awake, behaving animals. At the same time, new data pertaining to the structure of the cortex will be incorporated into the model as it emerges.

ACKNOWLEDGEMENTS

We wish to thank Dr. Lewis Haberly and Dr. Joshua Chover for their roles in the development and continued support of the modeling effort. We also wish to thank Dave Bilitch for his technical assistance. This work was supported by NIH grant NS22205, NSF grant EET-8700064, the Lockheed Corporation, and a fellowship from the ARCS foundation.

APPENDIX

Somatic Integration

$$\frac{dV_i}{dt} = \frac{1}{c_m}\left[\sum_{k=1}^{n_{types}} I_{ik}(t) + \frac{E_r - V_i(t)}{r_l}\right] \tag{1.0}$$

$$I_{ik}(t) = [E_k - V_i(t)]\,g_{ik}(t) \tag{1.1}$$

n_{types} = number of input types
$V_i(t)$ = membrane potential of ith cell
$I_{ik}(t)$ = current into cell i due to input type k
E_k = equilibrium potential associated with input type k
E_r = resting potential
r_l = membrane leakage resistance
c_m = membrane capacitance
$g_{ik}(t)$ = conductance due to input type k in cell i

Spike Propagation and Synaptic Input

$$g_{ik}(t) = \sum_{j=1}^{n_{cells}} \int_{\lambda=0}^{\lambda=d_k} F_k(\lambda)\, A_{ijk}\, W_{ij}\, S_j\!\left(t - \lambda - \frac{L_{ij}}{v_k} - \varepsilon_k\right) d\lambda \tag{2.0}$$

$$F_k(t) = \frac{t}{\tau} e^{\left(1 - \frac{t}{\tau}\right)} \left[(1 - U(t-\tau)) + U(t-\tau)\cos\left[\frac{\pi}{2}\frac{(t-\tau)}{(d_k-\tau)}\right]\right], \quad \tau = \gamma d_k \tag{2.1}$$

$$A_{ijk} = (1 - \rho_k^{min})e^{-L_{ij}\rho_k} + \rho_k^{min} \tag{2.2}$$

$$S_j(t) = \begin{cases} 1 & V_j(t) > T_j,\; S_j(\lambda) = 0 \text{ for } \lambda = t..t - \Delta t_r \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

$$L_{ij} = |i - j|\,\Delta x \tag{2.4}$$

n_{cells} = number of cells in the simulation
Δx = distance between adjacent cells
d_k = duration of conductance change due to input type k
v_k = velocity of signals for input type k
ε_k = latency for input type k
ρ_k = spatial attenuation factor for input type k
ρ_k^{min} = minimum spatial attenuation for input type k
Δt_r = refractory period
T_j = threshold for cell j
L_{ij} = distance from cell i to cell j
A_{ijk} = distribution of synaptic density for input type k
W_{ij} = synaptic weight from cell j to cell i
$g_{ik}(t)$ = conductance due to input type k in cell i
$F_k(t)$ = conductance waveform for input type k
$S_j(t)$ = spike output of cell j at time t
$U(t)$ = unit step function

Field Potentials

$$V_{ep}^{j}(t) = \frac{R_e}{4\pi}\sum_{i=1}^{n_{cells}}\sum_{n=1}^{n_{segs}} \frac{I_m^{in}(t)}{\left[(z_{elec} - z_n)^2 + (x_j - x_i)^2\right]^{\frac{1}{2}}} \tag{3.0}$$

n_{cells} = number of cells in the simulation
n_{segs} = number of segments in the compartmental model
$V_{ep}^{j}(t)$ = approximate extracellular field potential at cell j
$I_m^{in}(t)$ = membrane current for segment n in cell i
z_{elec} = depth of recording site
z_n = depth of segment n
x_j = x location of the jth cell
R_e = extracellular resistance per unit length

Dendritic Model

$$\frac{dV_n}{dt} = \frac{1}{c_m^n}\left[I_a^{n-}(t) + I_a^{n+}(t) + \frac{E_r - V_n(t)}{r_m^n} + \sum_{c=1}^{n_{chan}} [E_c - V_n(t)]\,g_{nc}(t)\right] \tag{4.0}$$

$$I_a^{n-}(t) = \frac{V_{n-1}(t) - V_n(t)}{r_a^{n-1} + r_a^n}, \quad I_a^{n+}(t) = \frac{V_{n+1}(t) - V_n(t)}{r_a^{n+1} + r_a^n} \tag{4.1}$$

$$I_m^n(t) = I_a^{n-}(t) + I_a^{n+}(t) \tag{4.2}$$

$$r_a^n = \frac{1}{2}\left[R_e l_n + R_i \frac{l_n}{\pi\left(\frac{d_n}{2}\right)^2}\right] \quad , \quad r_m^n = \frac{R_m}{\pi l_n d_n} \quad , \quad c_m^n = C_m \pi l_n d_n \tag{4.3}$$

n_{chan} = number of different channels per segment
$V_n(t)$ = membrane potential of nth segment
c_m^n = membrane capacitance for segment n
r_a^n = axial resistance for segment n
r_m^n = membrane resistance for segment n
$g_{nc}(t)$ = conductance of channel c in segment n
E_c = equilibrium potential associated with channel c
$I_a^{n\pm}(t)$ = axial current between segment $n\pm1$ and n

$I_m^n(t)$ = membrane current for segment n
l_n = length of segment n
d_n = diameter of segment n
R_m = membrane resistivity
R_i = intracellular resistivity per unit length
R_e = extracellular resistance per unit length
C_m = capacitance per unit surface area

REFERENCES

1. W. J. Freeman, J. Neurophysiol., 23, 111 (1960).
2. T. Tanabe, M. Iino, and S. F. Takagi, J. Neurophysiol., 38,1284 (1975).
3. L. B. Haberly, Chemical Senses, 10, 219 (1985).
4. M. Wilson, J. M. Bower, J. Chover, and L. B. Haberly, Soc. Neuro. Abs., 11, 317 (1986).
5. M. Wilson and J. M. Bower, Soc. Neurosci. Abs., 12, 310 (1987).
6. M. Devor, J. Comp. Neur., 166, 31 (1976).
7. L. B. Haberly and J. L. Price, J. Comp. Neurol., 178, 711 (1978a).
8. L. B. Haberly and S. Presto, J. Comp. Neur., 248, 464 (1986).
9. L. B. Haberly and J. M. Bower, J. Neurophysiol., 51, 90 (1984).
10. M. A. Biedenbach and C. F. Stevens, J. Neurophysiol., 32, 193 (1969).
11. M. A. Biedenbach and C. F. Stevens, J. Neurophysiol., 32, 204 (1969).
12. M. Satou, K. Mori, Y. Tazawa, and S. F. Takagi, J. Neurophysiol., 48, 1157 (1982).
13. G. F. Tseng and L. B. Haberly, Soc. Neurosci. Abs. 12, 667 (1986).
14. L. B. Luskin and J. L. Price, J. Comp. Neur., 216, 264 (1983).
15. J. M. Bower and L. B. Haberly,L.B., Proc. Natl. Acad. Sci. USA, 83, 1115 (1985).
16. W. J. Freeman, J. Neurophysiol., 31, 1 (1968).
17. L. B. Haberly, J. Neurophysiol., 36, 762 (1973).
18. L. B. Haberly, J. Neurophysiol., 36, 775 (1973).
19. W. J. Freeman, Electroenceph. and Clin. Neurophysiol., 44, 586 (1978).
20. W.J. Freeman and W. Schneider, Psychophysiology, 19, 44 (1982).
21. F. Macrides and S. L. Chorover, Science, 175, 84 (1972).
22. F. Macrides, H. B. Eigenbaum, and W. B. Forbes, J. Neurosci., 2, 12, 1705 (1982).
23. P. D. MacLean, N. H. Horwitz, and F. Robinson, Yale J. Biol. Med., 25, 159 (1952).
24. E. D. Adrian, Electroenceph. and Clin. Neurophysiol., 2, 377 (1950).
25. W. J. Freeman, Exp. Neurol., 10, 525 (1964).

OMIT

To appear in:
Neural Information Processing Systems
D. Anderson (ed.)
AIP Press

NEURAL NETWORKS FOR TEMPLATE MATCHING: APPLICATION TO REAL-TIME CLASSIFICATION OF THE ACTION POTENTIALS OF REAL NEURONS

Yiu-fai Wong†, Jashojiban Banik† and James M. Bower‡
†Division of Engineering and Applied Science
‡Division of Biology
California Institute of Technology
Pasadena, CA 91125

ABSTRACT

Much experimental study of real neural networks relies on the proper classification of extracellulary sampled neural signals (i.e. action potentials) recorded from the brains of experimental animals. In most neurophysiology laboratories this classification task is simplified by limiting investigations to single, electrically well-isolated neurons recorded one at a time. However, for those interested in sampling the activities of many single neurons simultaneously, waveform classification becomes a serious concern. In this paper we describe and constrast three approaches to this problem each designed not only to recognize isolated neural events, but also to separately classify temporally overlapping events in real time. First we present two formulations of waveform classification using a neural network template matching approach. These two formulations are then compared to a simple template matching implementation. Analysis with real neural signals reveals that simple template matching is a better solution to this problem than either neural network approach.

INTRODUCTION

For many years, neurobiologists have been studying the nervous system by using single electrodes to serially sample the electrical activity of single neurons in the brain. However, as physiologists and theorists have become more aware of the complex, nonlinear dynamics of these networks, it has become apparent that serial sampling strategies may not provide all the information necessary to understand functional organization. In addition, it will likely be necessary to develop new techniques which sample the activities of multiple neurons simultaneously[1]. Over the last several years, we have developed two different methods to acquire multineuron data. Our initial design involved the placement of many tiny microelectrodes individually in a tightly packed pseudo-floating configuration within the brain[2]. More recently we have been developing a more sophisticated approach which utilizes recent advances in silicon technology to fabricate multi-ported silicon based electrodes (Fig. 1). Using these electrodes we expect to be able to readily record the activity patterns of larger number of neurons.

As research in multi-single neuron recording techniques continue, it has become very clear that whatever technique is used to acquire neural signals from many brain locations, the technical difficulties associated with sampling, data compressing, storing, analyzing and interpreting these signals largely dwarf the development of the sampling device itself. In this report we specifically consider the need to assure that neural action potentials (also known as "spikes") on each of many parallel recording channels are correctly classified, which is just one aspect of the problem of post-processing multi-single neuron data. With more traditional single electrode/single neuron recordings, this task usually in-

volves passing analog signals through a Schmidt trigger whose output indicates the occurence of an event to a computer, at the same time as it triggers an oscilloscope sweep of the analog data. The experimenter visually monitors the oscilloscope to verify the accuracy of the discrimination as a well-discriminated signal from a single neuron will overlap on successive oscilloscope traces (Fig. 1c). Obviously this approach is impractical when large numbers of channels are recorded at the same time. Instead, it is necessary to automate this classification procedure. In this paper we will describe and contrast three approaches we have developed to do this.

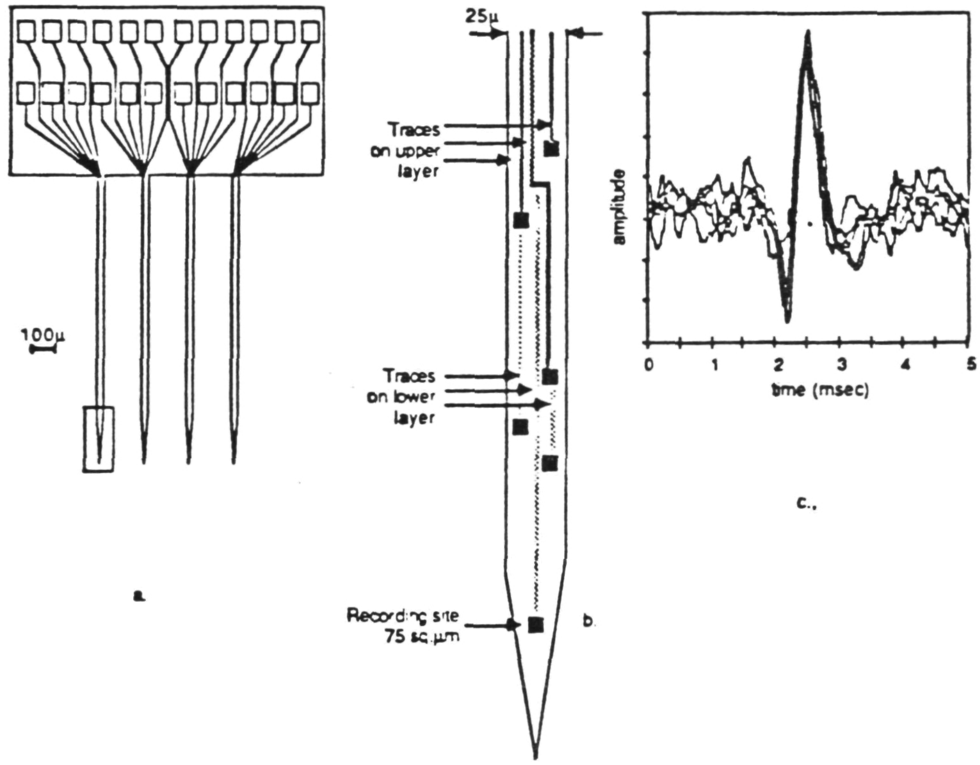


Fig. 1. Silicon probe being developed in our lababoratory for multi-single unit recording in cerebellar cortex. a) a complete probe; b) surface view of one recording tip; c) several superimposed neuronal action potentials recorded from such a silicon electrode in cerebellar cortex.

While our principal design objective is the assurance that neural waveforms are adequately discriminated on multiple channels, technically the overall objective of this research project is to sample from as many single neurons as possible. Therefore, it is a natural extention of our effort to develop a neural waveform classification scheme robust enough to allow us to distinguish activities arising from more than one neuron per recording site. To do this, however, we now not only have to determine that a particular signal is neural in origin, but also from which of several possible neurons it arose (see Fig. 2a). While in general signals from different neurons have different waveforms aiding in the classification, neurons recorded on the same channel firing simultaneously or nearly simultaneously will produce novel combination waveforms (Fig. 2b) which also need to be classified. It is this last complication which particularly

bedevils previous efforts to classify neural signals (For review see 5, also see 3-4). In summary, then, our objective was to design a circuit that would:

1. **distinguish different waveforms** even though neuronal discharges tend to be quite similar in shape (Fig. 2a);
2. **recognize the same waveform** even though unavoidable movements such as animal respiration often result in periodic changes in the amplitude of a recorded signal by moving the brain relative to the tip of the electrode;
3. be considerably **robust to recording noise** which variably corrupts all neural recordings (Fig. 2);
4. **resolve overlapping waveforms**, which are likely to be particularly interesting events from a neurobiological point of view;
5. provide **real-time performance** allowing the experimenter to detect problems with discrimination and monitor the progress of the experiment;
6. **be implementable in hardware** due to the need to classify neural signals on many channels simultaneously. Simply duplicating a software-based algorithm for each channel will not work, but rather, multiple, small, independent, and programmable hardware devices need to be constructed.

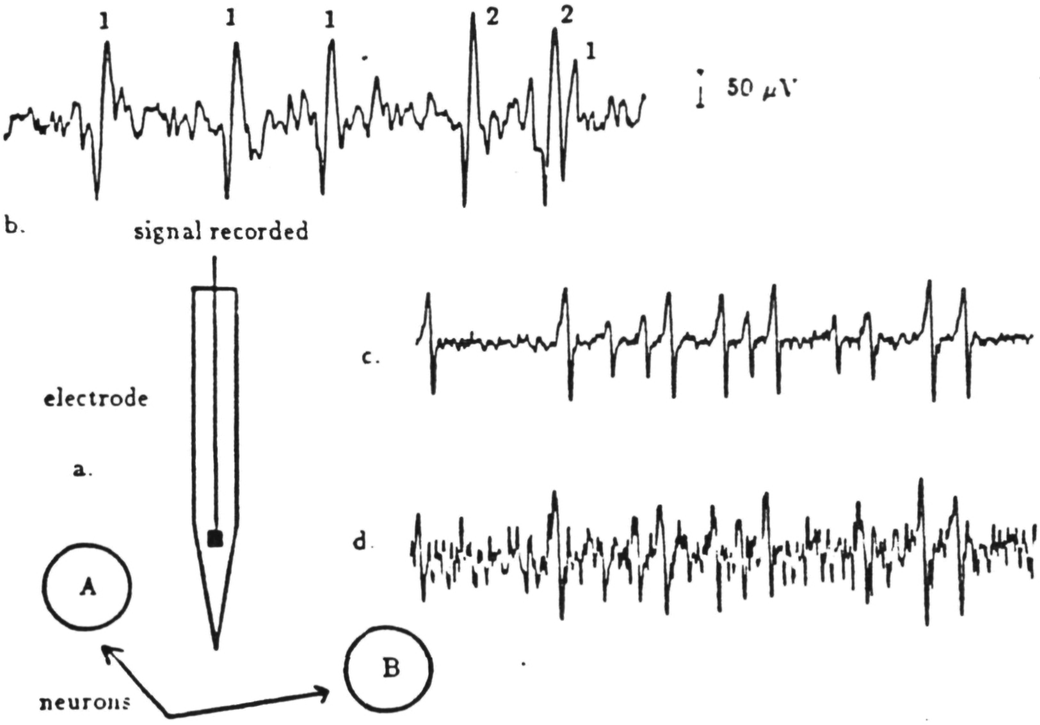


Fig. 2. a) Schematic diagram of an electrode recording from two neuronal cell bodies b) An actual multi-neuron recording. Note the similarities in the two waveforms and the overlapping event. c) and d) Synthesized data with different noise levels for testing classification algorithms (c: 0.3 NSR ; d: 1.1 NSR).

METHODS

The problem of detecting and classifying multiple neural signals on single voltage records involves two steps. First, the waveforms that are present in a particular signal must be identified and the templates be generated; second, these waveforms must be detected and classified in ongoing data records. To accomplish the first step we have modified the principal component analysis procedure described by Abeles and Goldstein[3] to automatically extract templates of the distinct waveforms found in an initial sample of the digitized analog data. This will not be discussed further as it is the means of accomplishing the second step which concerns us here. Specifically, in this paper we compare three new approaches to ongoing waveform classification which deal explicitly with overlapping spikes and variably meet other design criteria outlined above. These approaches consist of a modified template matching scheme, and two applied neural network implementations. We will first consider the neural network approaches. On a point of nomenclature, to avoid confusion in what follows, the real neurons whose signals we want to classify will be referred to as "neurons" while computing elements in the applied neural networks will be called "Hopons."

Neural Network Approach — Overall, the problem of classifying neural waveforms can best be seen as an optimization problem in the presence of noise. Much recent work on neural-type network algorithms has demonstrated that these networks work quite well on problems of this sort[6-8]. In particular, in a recent paper Hopfield and Tank describe an A/D converter network and suggest how to map the problem of template matching into a similar context[8]. The energy functional for the network they propose has the form:

$$E = \frac{-1}{2} \sum_i \sum_j T_{ij} V_i V_j - \sum_i V_i I_i \tag{1}$$

where T_{ij} = connectivity between Hopon i and Hopon j, V_i = voltage output of Hopon i, I_i = input current to Hopon i and each Hopon has a sigmoid input-output characteristic $V = g(u) = 1/(1 + exp(-au))$.

If the equation of motion is set to be:

$$du_i/dt = -\partial E/\partial V = \sum_j T_{ij} V_j + I_i \tag{1a}$$

then we see that $dE/dt = -\left(\sum_j T_{ij} V_j + I_i\right) dV/dt = -(du/dt)(dV/dt) = -g'(u)(du/dt)^2 \leq 0$. Hence E will go to to a minimum which, in a network constructed as described below, will correspond to a proposed solution to a particular waveform classification problem.

Template Matching using a Hopfield-type Neural Net — We have taken the following approach to template matching using a neural network. For simplicity, we initially restricted the classification problem to one involving two waveforms and have accordingly constructed a neural network made up of two groups of Hopons, each concerned with discriminating one or the other waveform. The classification procedure works as follows: first, a Schmidt trigger

is used to detect the presence of a voltage on the signal channel above a set threshold. When this threshold is crossed, implying the presence of a possible neural signal, 2 msecs of data around the crossing are stored in a buffer (40 samples at 20 KHz). Note that biophysical limitations assure that a single real neuron cannot discharge more than once in this time period, so only one waveform of a particular type can occur in this data sample. Also, action potentials are of the order of 1 msec in duration, so the 2 msec window will include the full signal for single or overlapped waveforms. In the next step (explained later) the data values are correlated and passed into a Hopfield network designed to minimize the mean-square error between the actual data and the linear combination of different delays of the templates. Each Hopon in the set of Hopons concerned with one waveform represents a particular temporal delay in the occurrence of that waveform in the buffer. To express the network in terms of an energy function formulation: Let $x(t)$ = input waveform amplitude in the t^{th} time bin, $s_j(t)$ = amplitude of the j^{th} template, V_{jk} denote if $s_j(t-k)$(j^{th} template delayed by k time bins)is present in the input waveform. Then the appropriate energy function is:

$$\begin{aligned} E = & \frac{1}{2}\sum_t\Big(x(t) - \sum_{j,k} V_{jk}s_j(t-k)\Big)^2 \\ & - \frac{1}{2}\sum_{t,j,k} V_{jk}(V_{jk}-1)s_j^2(t-k) \\ & + \gamma \sum_{j,k_1<k_2} V_{jk_1}V_{jk_2} \end{aligned} \tag{2}$$

The first term is designed to minimize the mean-square error and specifies the best match. Since $V \in [0,1]$, the second term is minimized only when each V_{jk} assumes values 0 or 1. It also sets the diagonal elements T_{ij} to 0. The third term creates mutual inhibition among the processing nodes evaluating the same neuronal signal, which as described above can only occur once per sample.

Expanding and simplifying expression (2), the connection matrix is:

$$T_{(j_1,k_1),(j_2,k_2)} = \begin{cases} -\sum_t s_{j_1}(t-k_1)s_{j_2}(t-k_2) - \gamma\delta_{j_1 j_2} & \\ 0 & \text{if } j_1 = j_2, k_1 = k_2 \end{cases} \tag{3a}$$

and the input current

$$I_{jk} = \sum_t x(t)s_j(t-k) - \frac{1}{2}\sum_t s_j^2(t-k) \tag{3b}$$

As it can be seen, the inputs are the correlations between the actual data and the various delays of the templates subtracting a constant term.

Modified Hopfield Network — As documented in more detail in Fig. 3-4, the above full Hopfield-type network works well for temporally isolated spikes at moderate noise levels, but for overlapping spikes it has a local minima problem. This is more severe with more than two waveforms in the network.

Further, we need to build our network in hardware and the full Hopfield network is difficult to implement with current technology (see below). For these reasons, we developed a modified neural network approach which significantly reduces the necessary hardware complexity and also has improved performance. To understand how this works, let us look at the information contained in the quantities T_{ij} and I_{ij} (eq. $3a$ and $3b$) and make some use of them. These quantities have to be calculated at a pre-processing stage before being loaded into the Hopfield network. If after calculating these quantities, we can quickly rule out a large number of possible template combinations, then we can significantly reduce the size of the problem and thus use a much smaller (and hence more efficient) neural network to find the optimal solution. To make the derivation simple, we define slightly modified versions of T_{ij} and I_{ij} (eq. $4a$ and $4b$) for two-template case.

$$T_{ij} = \sum_t s_1(t-i)s_2(t-j) \tag{4a}$$

$$I_{ij} = \sum_t x(t)\Big[\frac{1}{2}s_1(t-i) + \frac{1}{2}s_2(t-j)\Big] - \frac{1}{2}\sum_t s_1^2(t-i) - \frac{1}{2}\sum_t s_2^2(t-j) \tag{4b}$$

In the case of overlaping spikes the T_{ij}'s are the cross-correlations between $s_1(t)$ and $s_2(t)$ with different delays and I_{ij}'s are the cross-correlations between input $x(t)$ and weighted combination of $s_1(t)$ and $s_2(t)$. Now if $x(t) = s_1(t-i) + s_2(t-j)$ (i.e. the overlap of the first template with i time bin delay and the second template with j time bin delay), then $\Delta_{ij} = |T_{ij} - I_{ij}| = 0$. However in the presence of noise, Δ_{ij} will not be identically zero, but will equal to the noise, and if $\Delta_{ij} > \Delta T_{ij}$ (where $\Delta T_{ij} = |T_{ij} - T_{i'j'}|$ for $i \neq i'$ and $j \neq j'$) this simple algorithm may make unacceptable errors. A solution to this problem for overlapping spikes will be described below, but now let us consider the problem of classifying non-overlapping spikes. In this case, we can compare the input cross-correlation with the auto-correlations (eq. $4c$ and $4d$).

$$T_i' = \sum_t s_1^2(t-i); \quad T_i'' = \sum_t s_2^2(t-i) \tag{4c}$$

$$I_i' = \sum_t x(t)s_1(t-i); \quad I_i'' = \sum_t x(t)s_2(t-i) \tag{4d}$$

So for non-overlapping cases, if $x(t) = s_1(t-i)$, then $\Delta_i' = |T_i' - I_i'| = 0$. If $x(t) = s_2(t-i)$, then $\Delta_i'' = |T_i'' - I_i''| = 0$.

In the absence of noise, then the minimum of Δ_{ij}, Δ_i' and Δ_i'' represents the correct classification. However, in the presence of noise, none of these quantities will be identically zero, but will equal the noise in the input $x(t)$ which will give rise to unacceptible errors. Our solution to this noise related problem is to choose a few minima (three have chosen in our case) instead of one. For each minimum there is either a known corresponding linear combination of templates for overlapping cases or a simple template for non-overlapping cases. A three neuron Hopfield-type network is then programmed so that each neuron corresponds to each of the cases. The input $x(t)$ is fed to this tiny network to resolve whatever confusion remains after the first step of "cross-correlation" comparisons. (Note: Simple template matching as described below can also be used in the place of the tiny Hopfield type network.)

Simple Template Matching — To evaluate the performances of these neural network approaches, we decided to implement a simple template matching scheme, which we will now describe. However, as documented below, this approach turned out to be the most accurate and require the least complex hardware of any of the three approaches. The first step is, again, to fill a buffer with data based on the detection of a possible neural signal. Then we calculate the difference between the recorded waveform and all possible combinations of the two previously identified templates. Formally, this consists of calculating the distances between the input $x(m)$ and all possible cases generated by all the combinations of the two templates.

$$d_{ij} = \sum_t |x(t) - \{s_1(t-i) + s_2(t-j)\}|$$

$$d'_i = \sum_t |x(t) - s_1(t-i)|; \quad d''_i = \sum_t |x(t) - s_2(t-i)|$$

$$d_{min} = min(d_{ij}, d'_i, d''_i)$$

d_{min} gives the best fit of all possible combinations of templates to the actual voltage signal.

TESTING PROCEDURES

To compare the performance of each of the three approaches, we devised a common set of test data using the following procedures. First, we used the principal component method of Abeles and Goldstein[3] to generate two templates from a digitized analog record of neural activity recorded in the cerebellum of the rat. The two actual spike waveform templates we decided to use had a peak-to-peak ratio of 1.375. From a second set of analog recordings made from a site in the cerebellum in which no action potential events were evident, we determined the spectral characteristics of the recording noise. These two components derived from real neural recordings were then digitally combined, the objective being to construct realistic records, while also knowing absolutely what the correct solution to the template matching problem was for each occurring spike. As shown in Fig. 2c and 2d, data sets corresponding to different noise to signal ratios were constructed. We also carried out simulations with the amplitudes of the templates themselves varied in the synthesized records to simulate waveform changes due to brain movements often seen in real recordings. In addition to two waveform test sets, we also constructed three waveform sets by generating a third template that was the average of the first two templates. To further quantify the comparisons of the three diffferent approaches described above we considered non-overlapping and overlapping spikes separately. To quantify the performance of the three different approaches, two standards for classification were devised. In the first and hardest case, to be judged a correct classification, the precise order and timing of two waveforms had to be reconstructed. In the second and looser scheme, classification was judged correct if the order of two waveforms was correct but timing was allowed to vary by ± 100 μsecs(i.e. ± 2 time bins) which for most neurobiological applications is probably sufficient resolution. Figs. 3-4 compare the performance results for the three approaches to waveform classification implemented as digital simulations.

PERFORMANCE COMPARISON

Two templates - non-overlapping waveforms: As shown in Fig. 3a, at low noise-to-signal ratios (NSRs below .2) each of the three approaches were comparable in performance reaching close to 100% accuracy for each criterion. As the ratio was increased, however the neural network implementations did less and less well with respect to the simple template matching algorithm with the full Hopfield type network doing considerably worse than the modified network. In the range of NSR most often found in real data (.2 - .4) simple template matching performed considerably better than either of the neural network approaches. Also it is to be noted that simple template matching gives an estimate of the goodness of fit betwwen the waveform and the closest template which could be used to identify events that should not be classified (e.g. signals due to noise).

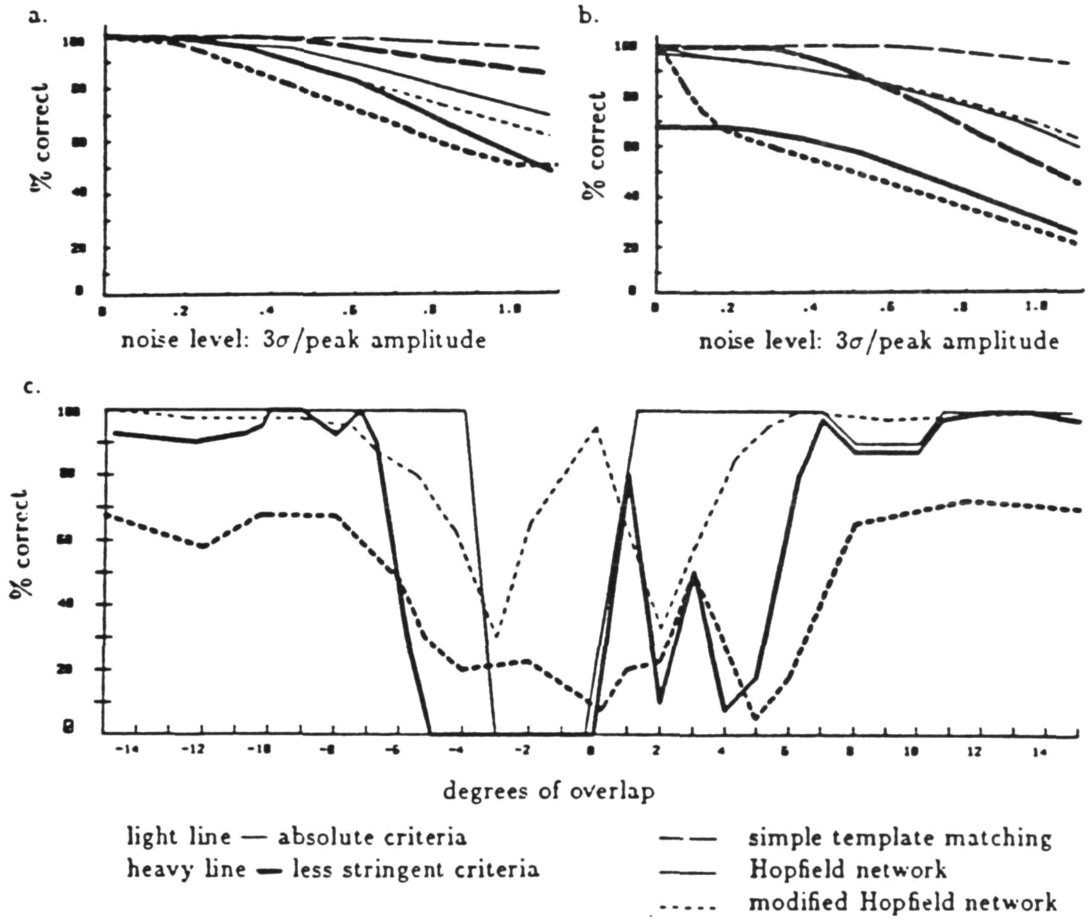


Fig. 3. Comparisons of the three approaches detecting two non-overlapping (a), and overlapping (b) waveforms, c) compares the performances of the neural network approaches for different degrees of waveform overlap.

Two templates - overlapping waveforms: Fig. 3b and 3c compare performances when waveforms overlapped. In Fig. 3b the serious local minima problem encountered in the full neural network is demonstrated as is the improved performance of the modified network. Again, overall performance in physi-

ological ranges of noise is clearly best for simple template matching. When the noise level is low, the modified approach is the better of the two neural networks due to the reliability of the correlation number which reflects the resemblence between the input data and the template. When the noise level is high, errors in the correlation numbers may exclude the right combination from the smaller network. In this case its performance is actually a little worse than the larger Hopfield network. Fig. 3c documents in detail which degrees of overlap produce the most trouble for the neural network approaches at average NSR levels found in real neural data. It can be seen that for the neural networks, the most serious problem is encountered when the delays between the two waveforms are small enough that the resulting waveform looks like the larger waveform with some perturbation.

Three templates – overlapping and non-overlapping: In Fig. 4 are shown the comparisons between the full Hopfield network approach and the simple template matching approach. For nonoverlapping waveforms, the performance of these two approaches is much more comparable than for the two waveform case (Fig. 4a), although simple template matching is still the optimal method. In the overlapping waveform condition, however, the neural network approach fails badly (Fig. 4b and 4c). For this particular application and implementation, the neural network approach does not scale well.

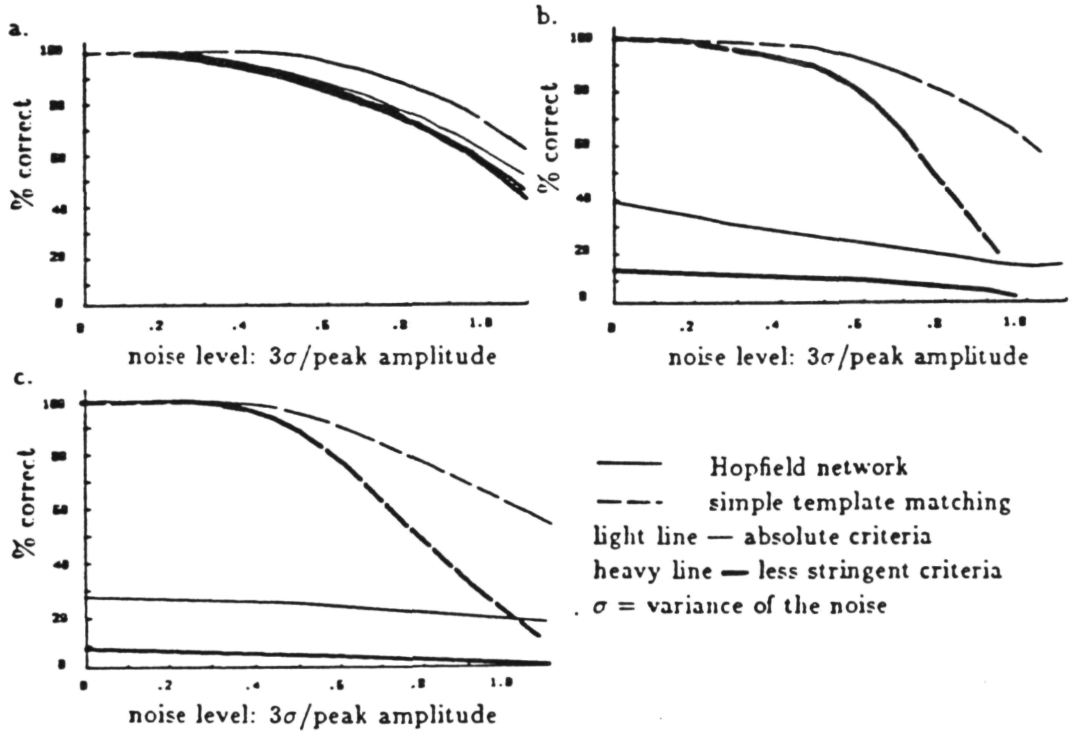


Fig. 4. Comparisons of performance for three waveforms. a) nonoverlapping waveforms; b) two waveforms overlapping; c) three waveforms overlapping.

HARDWARE COMPARISONS

As described earlier, an important design requirement for this work was the ability to detect neural signals in analog records in real-time originating from

many simultaneously active sampling electrodes. Because it is not feasible to run the algorithms in a computer in real time for all the channels simultaneously, it is necessary to design and build dedicated hardware for each channel. To do this, we have decided to design VLSI implementations of our circuitry. In this regard, it is well recognized that large modifiable neural networks need very elaborate hardware implementations. Let us consider, for example, implementing hardwares for a two-template case for comparisons. Let $n =$ no. of neurons per template (one neuron for each delay of the template), $m =$ no. of iterations to reach the stable state (in simulating the discretized differential equation, with step size $= 0.05$), $l =$ no. of samples in a template $t_j(m)$. Then, the number of connections in the full Hopfield network will be $4n^2$. The total no. of synaptic calculations $= 4mn^2$. So, for two templates and $n = 16, m = 100, 4mn^2 = 102,400$. Thus building the full Hopfield-type network digitally requires a system too large to be put in a single VLSI chip which will work in real time. If we want to build an analog system, we need to have many $(O(4n^2))$ easily modifiable synapses. As yet this technology is not available for nets of this size. The modified Hopfield-type network on the other hand is less technically demanding. To do the preprocessing to obtain the minimum values we have to do about $n^2 = 256$ additions to find all possible $I_{ij}s$ and require 256 subtractions and comparisons to find three minima. The costs associated with doing input cross-correlations are the same as for the full neural network (i.e. $2nl = 768 (l = 24)$ multiplications). The saving with the modified approach is that the network used is small and fast (120 multiplications and 120 additions to construct the modifiable synapses, no. of synaptic calculations $= 90$ with $m = 10, n = 3$).

In contrast to the neural networks, simple template matching is simple indeed. For example, it must perform about $n^2 l + n^2 = 10,496$ additions and $n^2 = 256$ comparisons to find the minimum d_{ij}. Additions are considerably less costly in time and hardware than multiplications. In fact, because this method needs only addition operations, our preliminary design work suggests it can be built on a single chip and will be able to do the two-template classification in as little as 20 microseconds. This actually raises the possibility that with switching and buffering one chip might be able to service more than one channel in essentially real time.

CONCLUSIONS

Template matching using a full Hopfield-type neural network is found to be robust to noise and changes in signal waveform for the two neural waveform classification problem. However, for a three-waveform case, the network does not perform well. Further, the network requires many modifiable connections and therefore results in an elaborate hardware implementation. The overall performance of the modified neural network approach is better than the full Hopfield network approach. The computation has been reduced largly and the hardware requirements are considerably less demanding demonstrating the value of designing a specific network to a specified problem. However, even the modified neural network performs less well than a simple template-matching algorithm which also has the simplest hardware implementation. Using the simple template matching algorithm, our simulations suggest it will be possible to build a two or three waveform classifier on a single VLSI chip using CMOS technology that works in real time with excellent error characteristics. Further, such a chip will be able to accurately classify variably overlapping

neural signals.

REFERENCES

[1] G. L. Gerstein, M. J. Bloom, I. E. Espinosa, S. Evanczuk & M. R. Turner, IEEE Trans. Sys. Cyb. Man., SMC-13, 668(1983).
[2] J. M. Bower & R. Llinas, Soc. Neurosci. Abst., 9, 607(1983).
[3] M. Abeles & M. H. Goldstein, Proc. IEEE, 65, 762(1977).
[4] W. M. Roberts & D. K. Hartline, Brain Res., 94, 141(1976).
[5] E. M. Schmidt, J. of Neurosci. Methods, 12, 95(1984).
[6] J. J. Hopfield, Proc. Natl. Acad. Sci.(USA), 81, 3088(1984).
[7] J. J. Hopfield & D. W. Tank, Biol. Cybern., 52, 141(1985).
[8] D. W. Tank & J. J. Hopfield, IEEE Trans. Circuits Syst., CAS-33, 533(1986).

ACKNOWLEDGEMENTS

We would like to acknowledge the contribution of Dr. Mark Nelson to the intellectual development of these projects and the able assistance of Herb Adams, Mike Walshe and John Powers in designing and constructing support equipment. This work was supported by NIH grant NS22205, the Whitaker Foundation and the Joseph Drown Foundation.

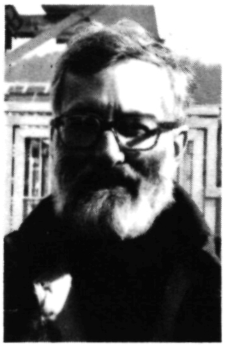
NOTES

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

511-61
15311

N91-71361

P.21



Walter J. Freeman, M.D.

University of California
at Berkeley
Berkeley, California

Dr. Freeman received his M.D. degree from Yale University (1954), clinical training at Johns Hopkins University, and postdoctoral training in neuroscience at the University of California at Los Angeles. He is a professor of physiology in the Department of Molecular and Cellular Biology at the University of California at Berkeley. Dr. Freeman has published, extensively, articles on linear and nonlinear neurodynamics of sensory systems, including a monograph (Mass Action in the Nervous System, Academic, 1975).

IMPLEMENTATION OF PATTERN-RECOGNITION ALGORITHMS DERIVED FROM OLFACTORY INFORMATION PROCESSING

Abstract

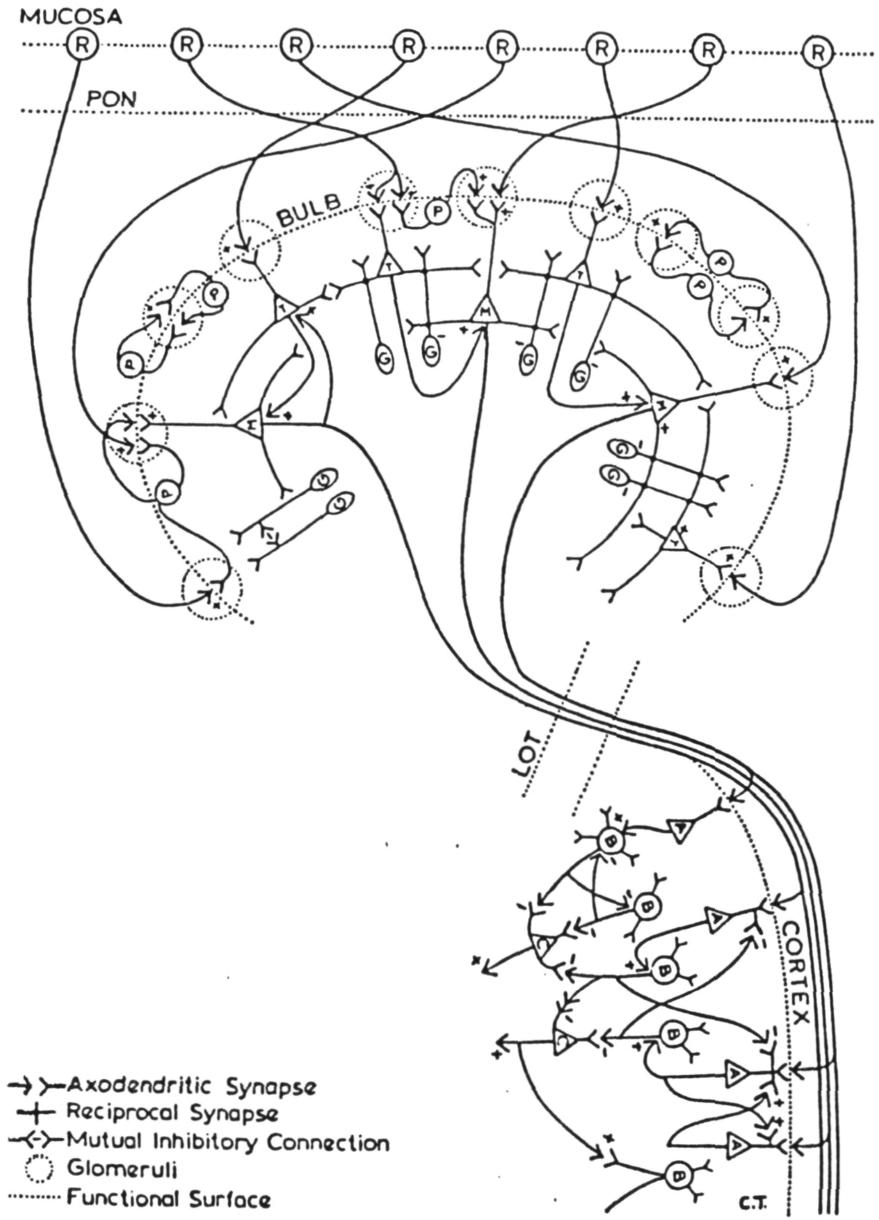
Sensory and perceptual information exists as space-time patterns of neural activity in cortex in two modes. Neural analysis of sensory input, as in feature extraction, is done with action potentials of single neurons in point processes. Neural synthesis of input with past experience and expectancy of future action is done with dendritic integration in local mean fields. Both kinds of activity are found to coexist in olfactory and visual cortex, each preceding and then following the other. The transformation of information from the pulse mode to the dendritic mode involves a state transition of the cortical network that can be modeled by a Hopf bifurcation in both software and hardware embodiments. These models show robust powers for amplification and correct classification of noisy and incomplete patterns corresponding to sensory inputs to biological nervous systems in attentive and motivated animals. The evidence is reviewed and the requirements are summarized for machine simulations of these operations.

C-3

IMPLEMENTATION OF PATTERN RECOGNITION ALGORITHMS DERIVED FROM OLFACTORY INFORMATION PROCESSING

SUMMARY

1. Modes of information in cerebral cortex -
    - point process: action potential frequency
    - local mean field: dendritic potential amplitude
2. Spatial amplitude modulation of carrier waves -
    - olfactory bulb of rabbit
    - primary visual cortex of monkey
3. Implementation with high-dimensional nonlinear ODEs
    - linear integration
    - asymmetric sigmoid nonlinearity
    - modifiable associational connections
4. Comparison of software and hardware embodiments
    - amplification and classification
    - chaos and the tolerance of disorder



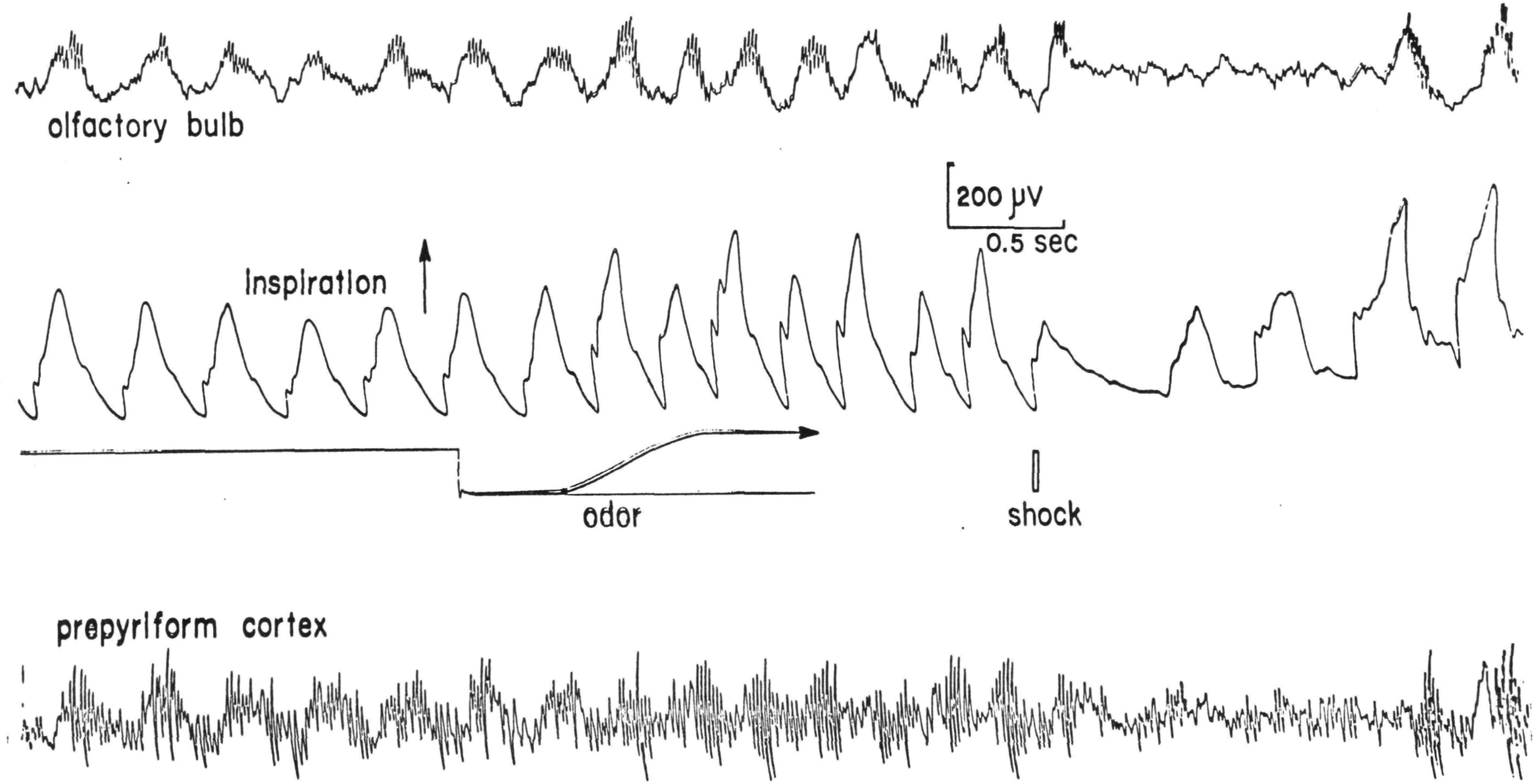
The main cell types in the olfactory bulb are the mitral and granule cells.

Mitral cells form a set of densely interconnected mutually excitatory cells. They also excite the granule cells. Mitral cell axons carry the output signal to the rest of the brain

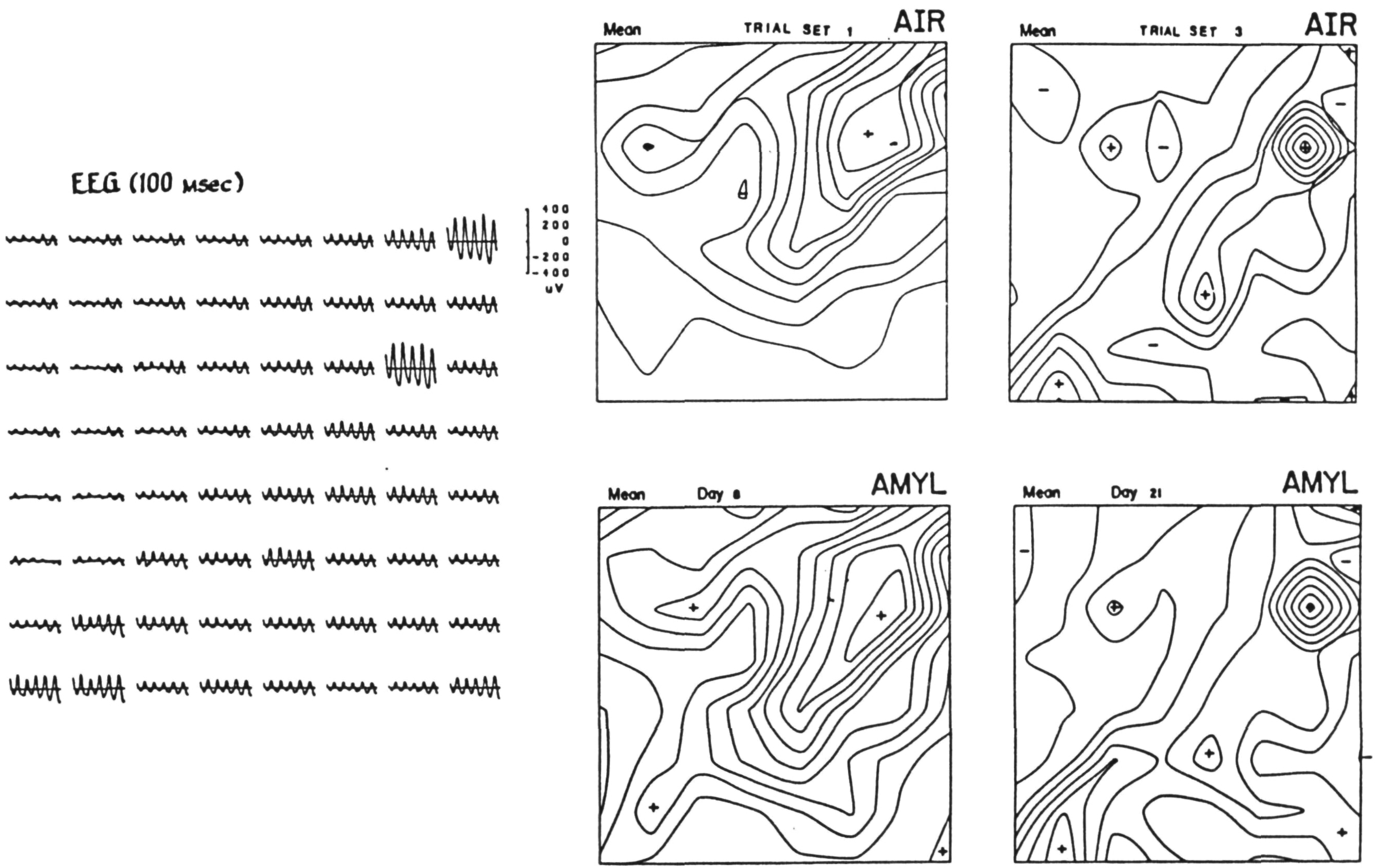
Granule cells form a set of densely interconnected mutually inhibitory neurons They also inhibit the mitral cells.

These two cell types are connected in a negative feedback loop. They form a neural oscillator. The olfactory bulb consists of approx. 2000 such coupled oscillators.

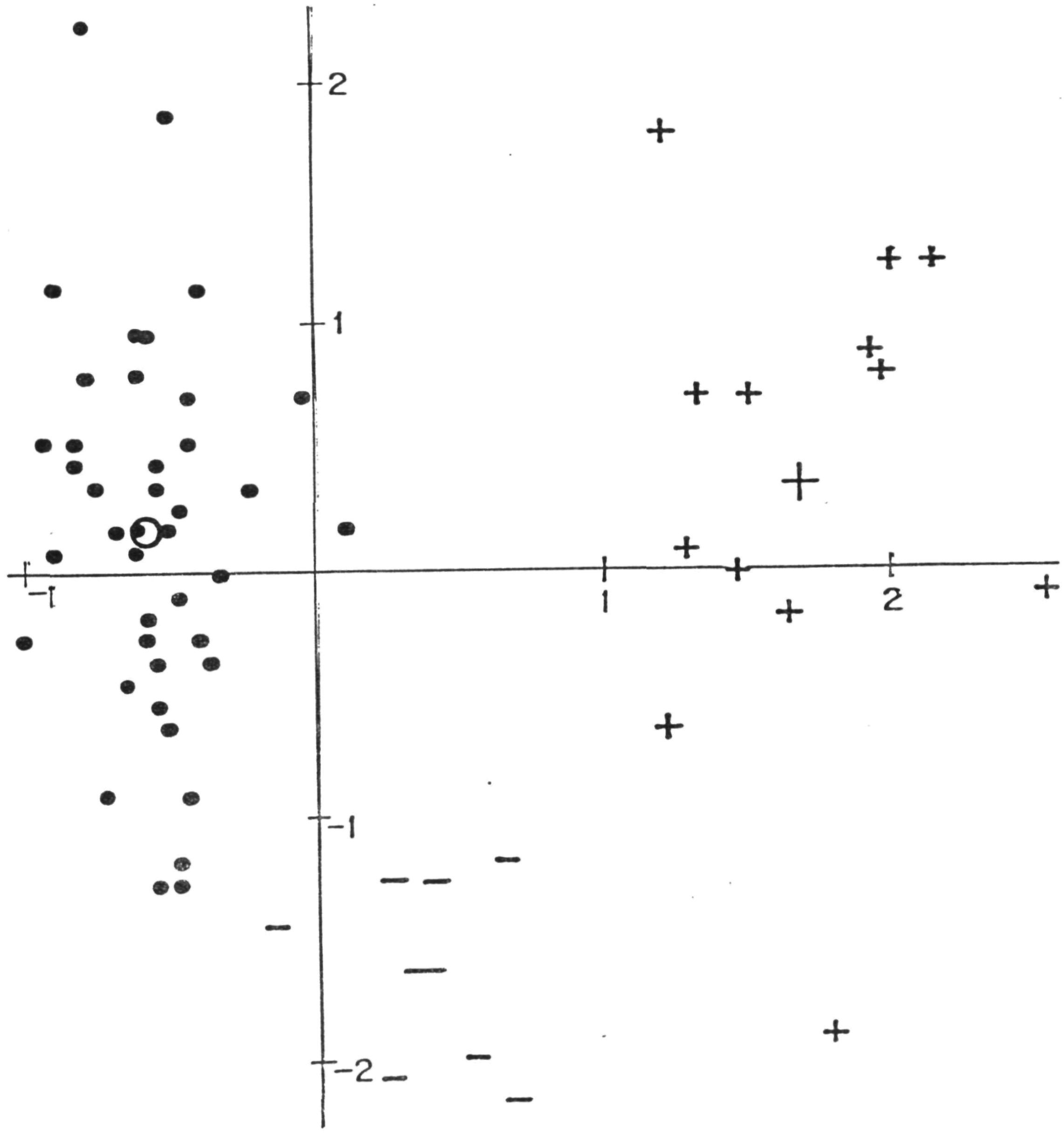
Excitatory couplings provide modifiable synapses in learning and perception. In hibitory couplings provide stability and spatial contrast.



THE BULB GENERATES A BRIEF OSCILLATORY EEG BURST WITH EACH INSPIRATION, WHETHER OR NOT A CONDITIONED STIMULUS ODOR IS PRESENTED. THIS RECORDING IS FROM ONE TRIAL IN A WAKING RABBIT. THE SAME PATTERN OF EEG IS FOUND OVER THE WHOLE MAIN BULB.

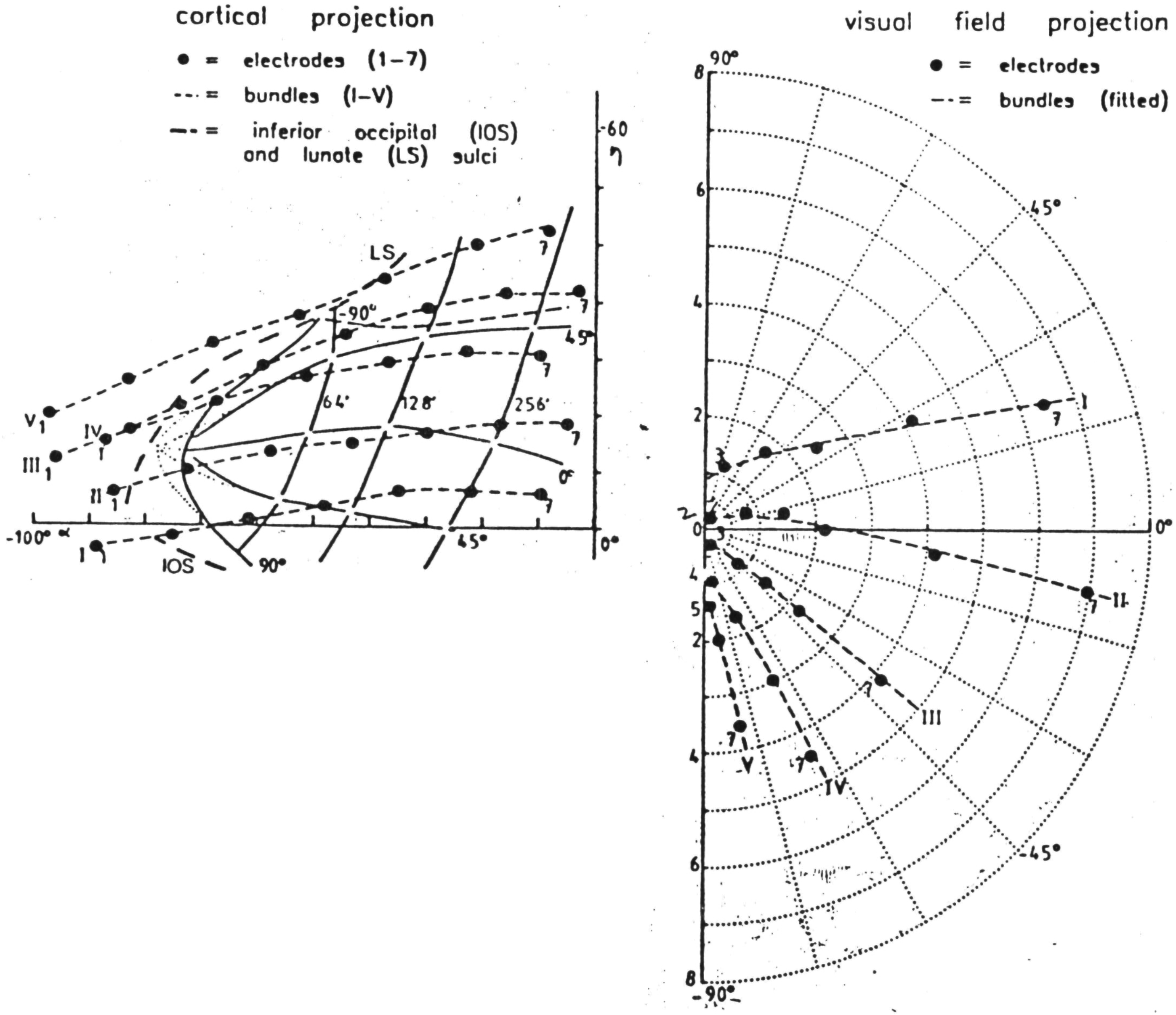


THE SPATIAL PATTERN OF ROOT MEAN SQUARE (RMS) AMPLITUDE FROM 64 EPIDURAL ELECTRODES (4 X 4 MM) CHANGES UNDER CONDITIONING AND RESTABILIZES AFTER EACH NEW TRAINING ODOR.



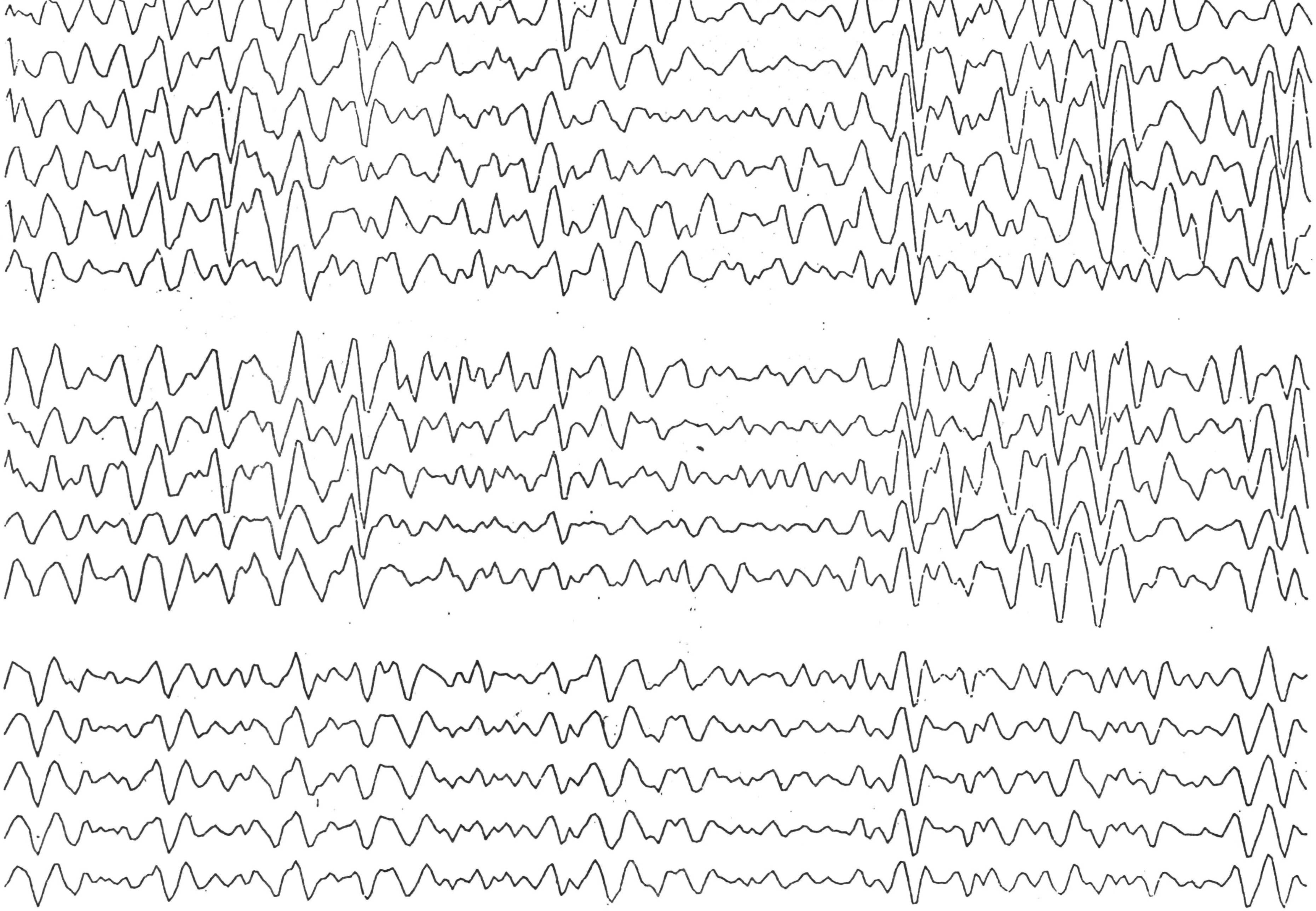
DISCRIMINANT ANALYSIS OF THE FACTOR SCORES SHOWS THAT 75% OF BURSTS ARE CORRECTLY CLASSIFIED WITH 2 DISCRIMINANT FUNCTIONS. A PLOT IS SHOWN OF THE DISCRIMINANT SPACE FOR ONE RABBIT.

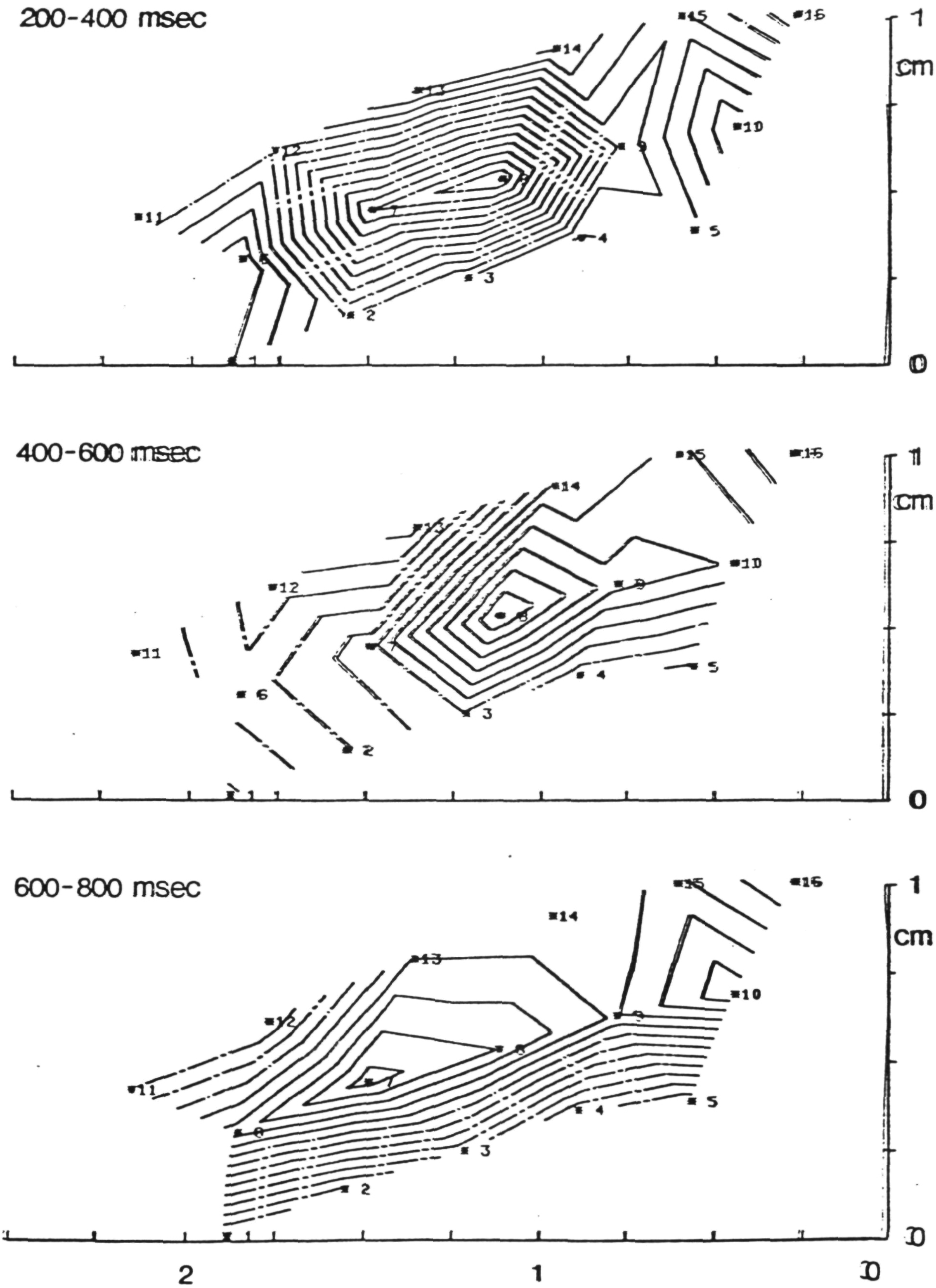
C



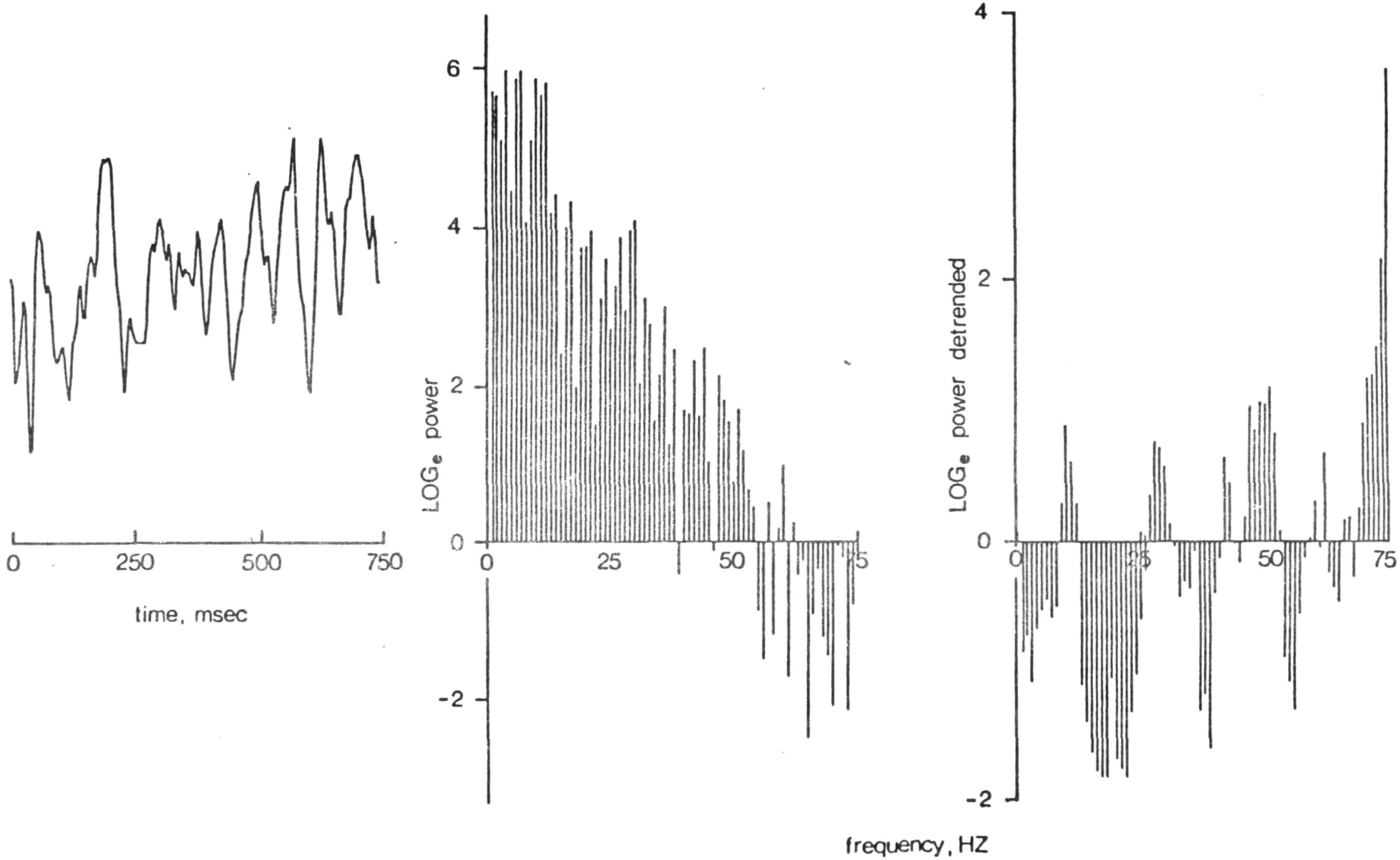
cortical projection
● = electrodes (1–7)
---= bundles (I–V)
—-= inferior occipital (IOS) and lunate (LS) sulci
LS
IOS
-90°
45°
64°
128°
256°
0°
90°
-100°
-60
visual field projection
● = electrodes
—-= bundles (fitted)
90°
45°
0°
-45°
-90°
I
II
III
IV
V

SCALE: 0080 uV/cm, 0050 ms/cm ref= REF files: FCWJ11, CORRECT RESPONSE





200-400 msec
400-600 msec
600-800 msec
cm
cm
cm
1
0
2
1
0
cm



0
250
500
750
time, msec
6
4
2
0
-2
LOG$_e$ power
0
25
50
75
4
2
0
-2
LOG$_e$ power detrended
0
25
50
75
frequency, HZ

BASIC ELEMENTS:

1. Linear integrator- 2nd order.
2. Static sigmoid nonlinearity.
3. Hebb connection & assembly.
4. Parallel input & output.

KEY PROPERTIES:

1. Chaotic basal state.
2. Input-dependent gain.
3. Bifurcation on input.
4. Spatial pattern coding.

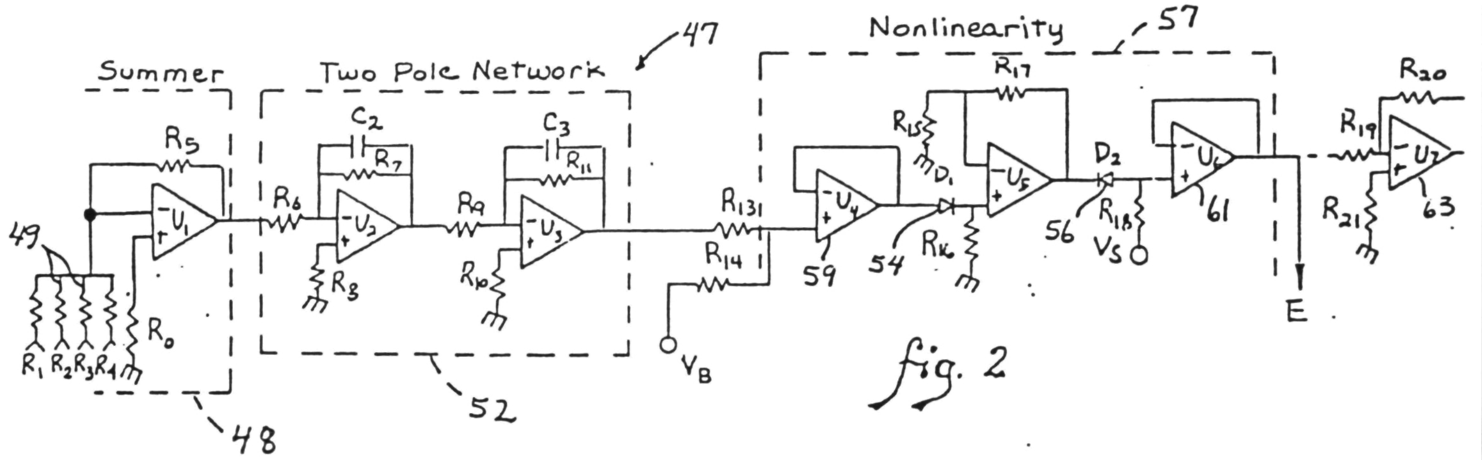
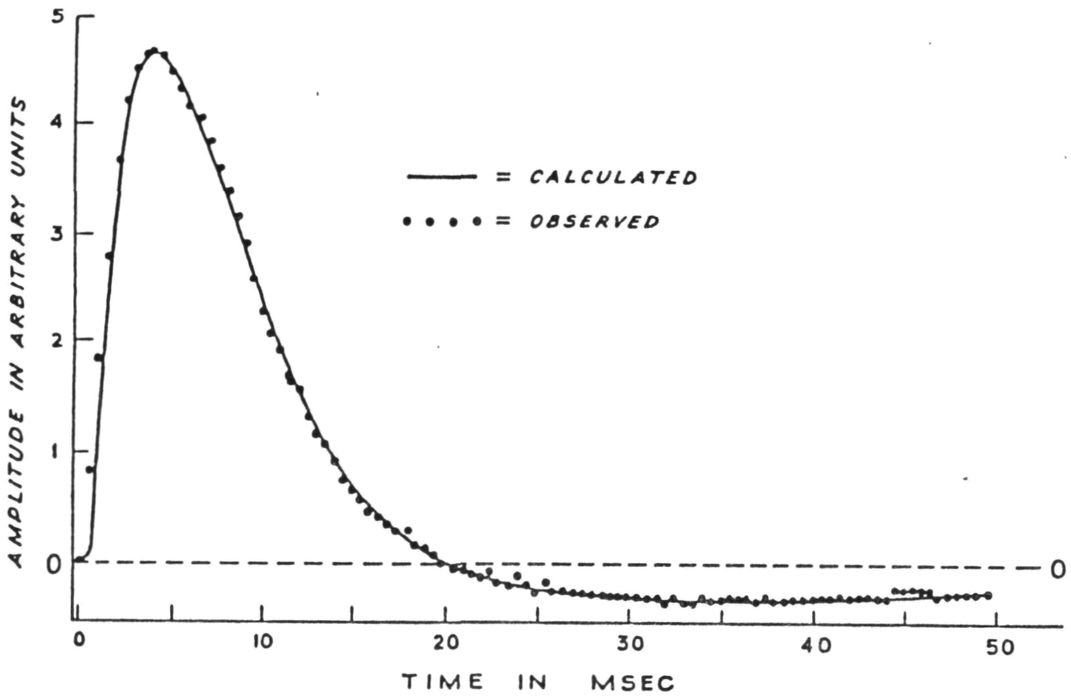
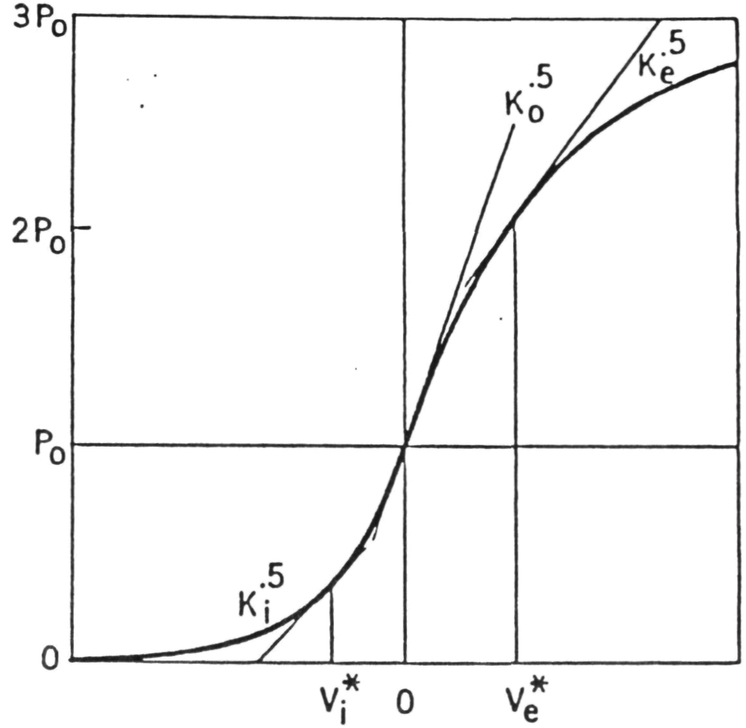


fig. 2





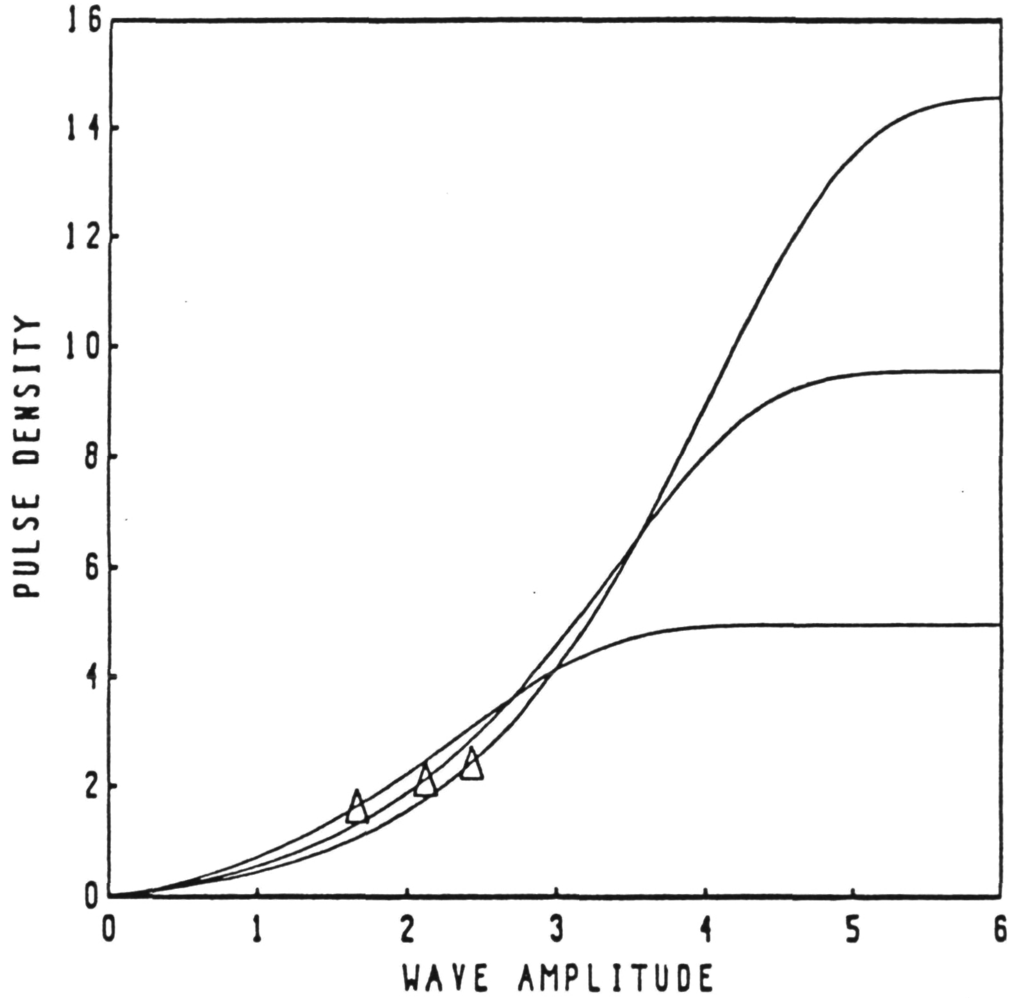
$$F(v_n) \triangleq \frac{1}{ab}\frac{d^2}{dt^2}\left[v_n(t)\right] + \frac{a+b}{ab}\frac{d}{dt}\left[v_n(t)\right] + v_n(t)$$

$$F(v_{e1,j}) = \zeta_e^j k_{ee}^{jj} p_{e2,j} - \zeta_e^j k_{ie}(p_{e1,j} + p_{i2,j}) + \sum_{k\neq j}^{N} \zeta_e^j k_{ee}^{jk} p_{e1,k} + I_j$$

$$F(v_{e2,j}) = \zeta_e^j k_{ee}^{jj} p_{e1,j} - \zeta_i k_{ie} p_{i1,j}$$

$$F(v_{i2,j}) = \zeta_e^j k_{ei} p_{e1,j} - \zeta_i k_{ii} p_{i1,j}$$

$$F(v_{i1,j}) = \zeta_e^j k_{ei}(p_{e1,j} + p_{e2,j}) - \zeta_i k_{ii} p_{i2,j} - \zeta_i k_{ii} \sum_{k\neq j}^{N} p_{i1,k}$$

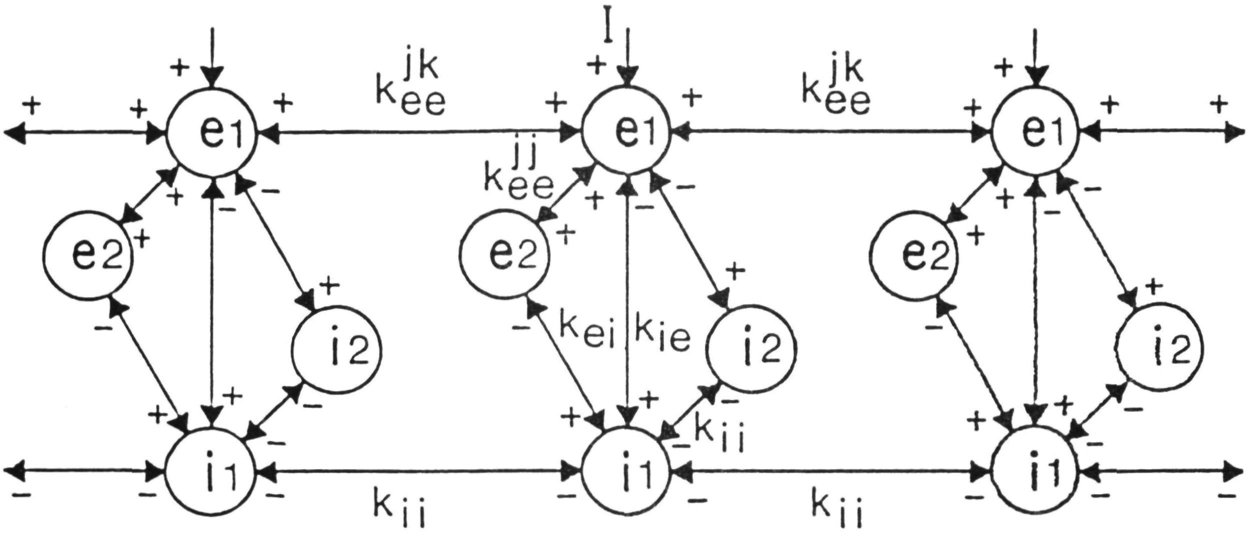


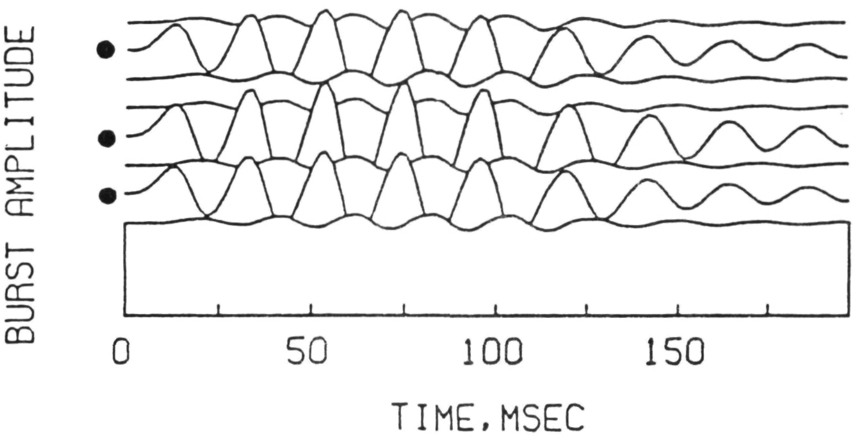
$$Q = Q_m \left\{ 1 - \exp\left[-(e^{v} - 1)/Q_m \right] \right\}, \quad v > -\mu_o$$

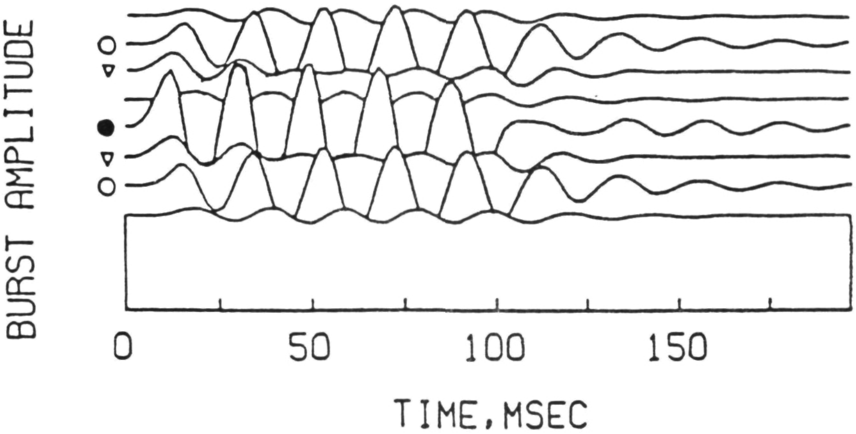
$$Q = -1, \quad v \leqslant \mu_o$$

$$\mu_o = -\ln\left[1 - Q_m \ln(1 + 1/Q_m) \right]$$

$$p = \mu_o (Q + 1)$$


I
k_{ee}^{jk}
k_{ee}^{jk}
k_{ee}^{jj}
k_{ei}
k_{ie}
k_{ii}
k_{ii}
k_{ii}
e1
e2
i1
i2

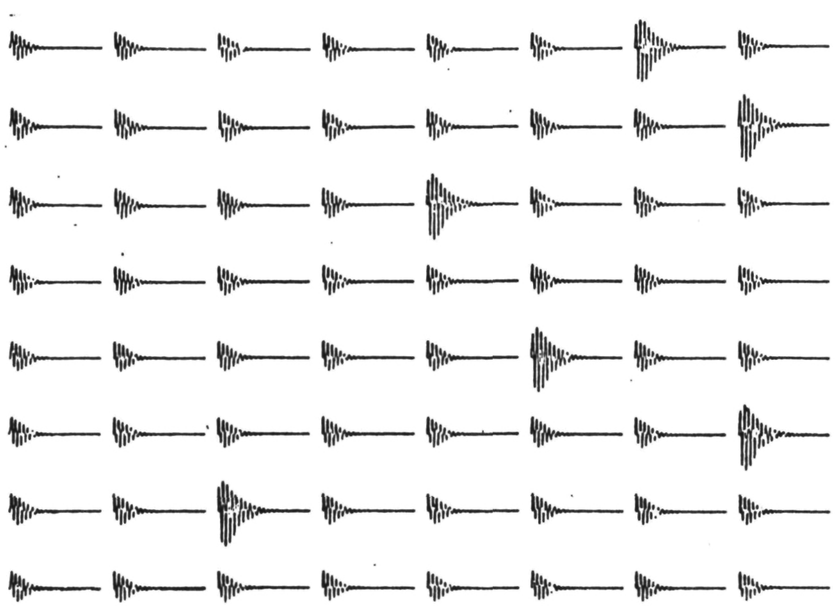

BURST AMPLITUDE
0
50
100
150
TIME, MSEC

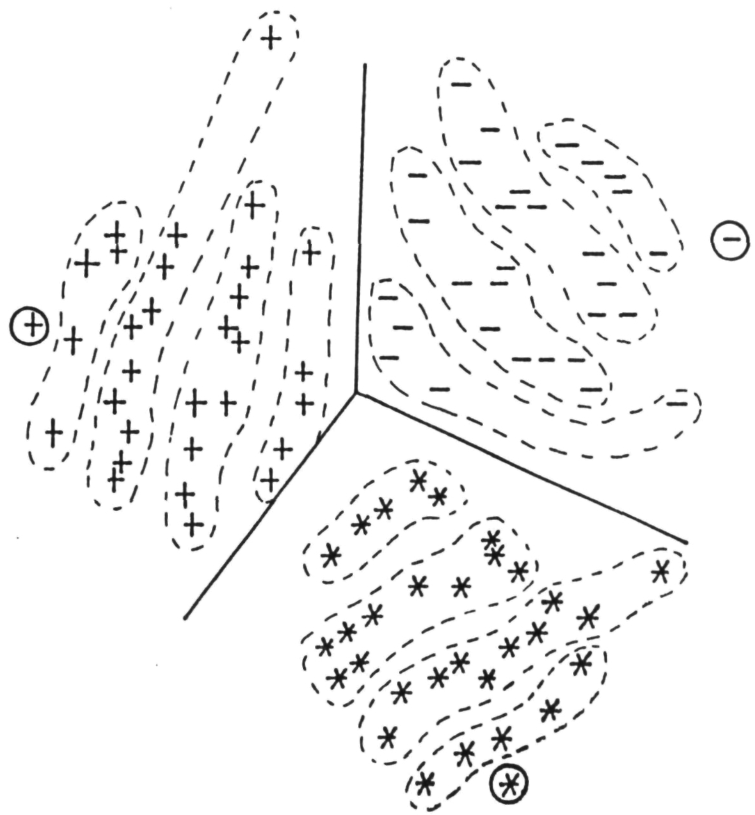

BURST AMPLITUDE
0
50
100
150
TIME, MSEC

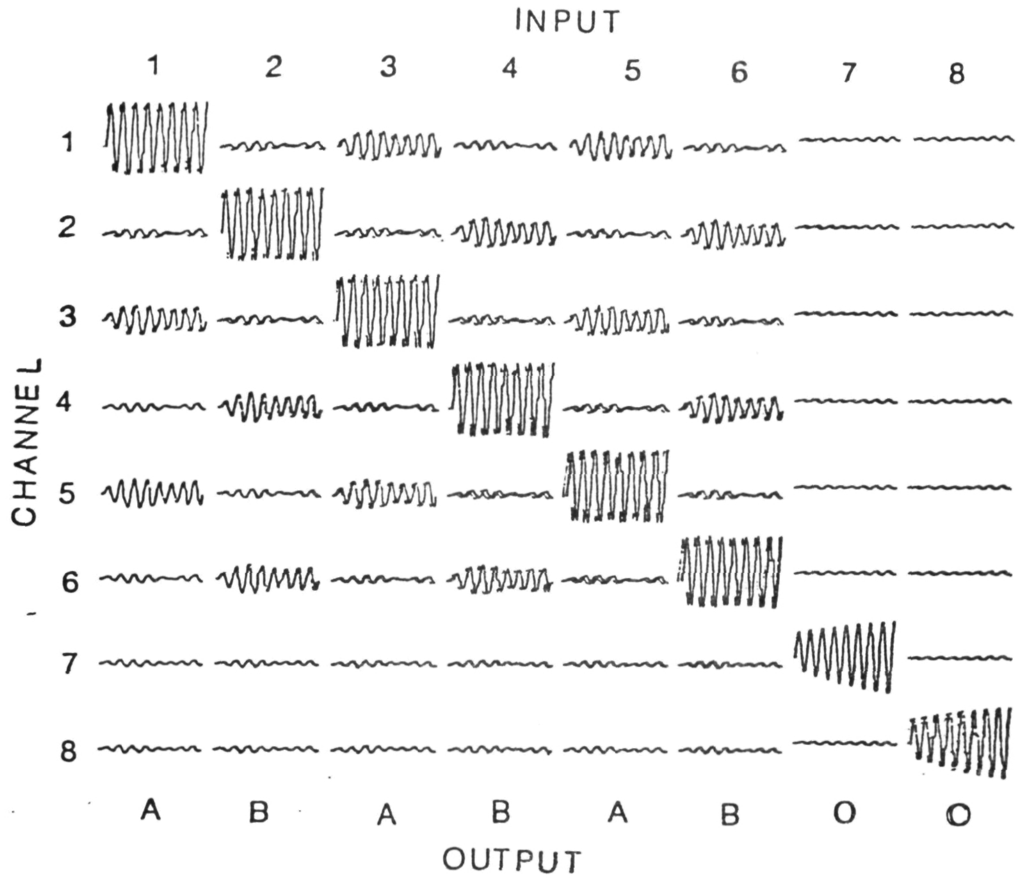
CORRELATION LEARNING RULE : a modified Hebb rule

changing synapse in the following way:
if both channel j and k are on, Kee(j,k)= Kee(high), otherwise, unchanged.
The output waveform in the 64-channel case

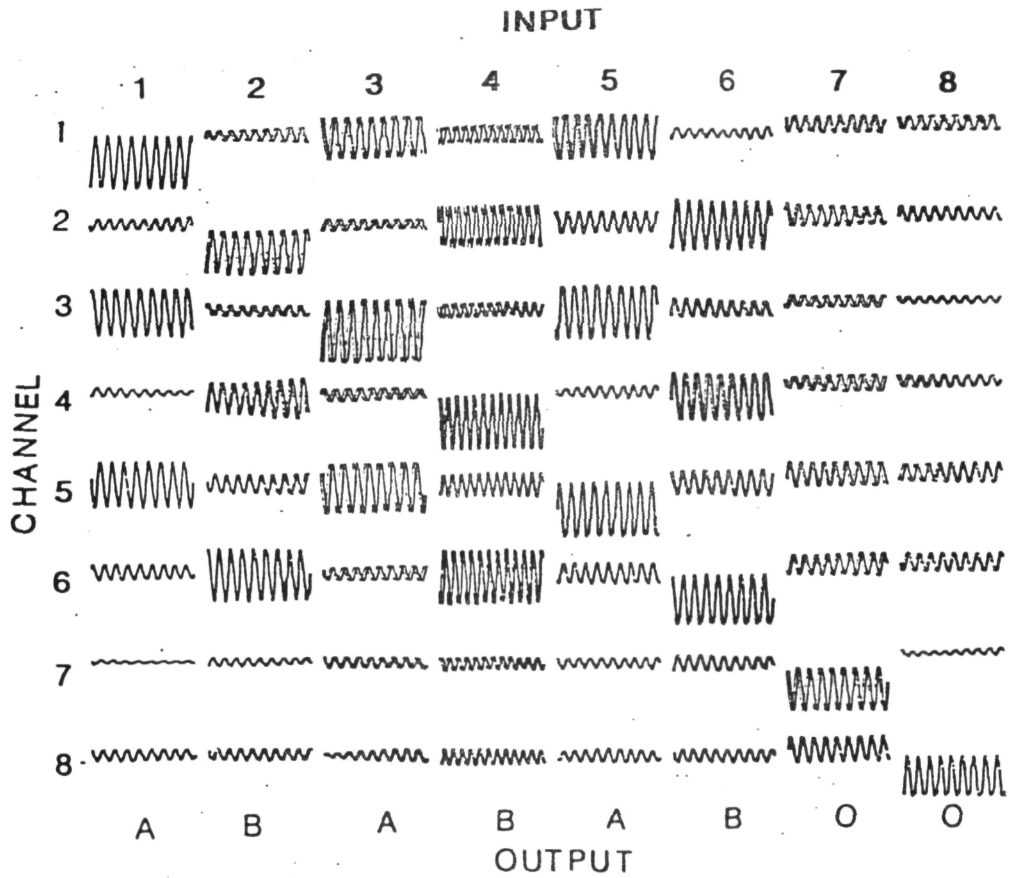
For the reduced interconnected KII set with 64-channel, the figure shows 100% clustering as well as perfect grouping.



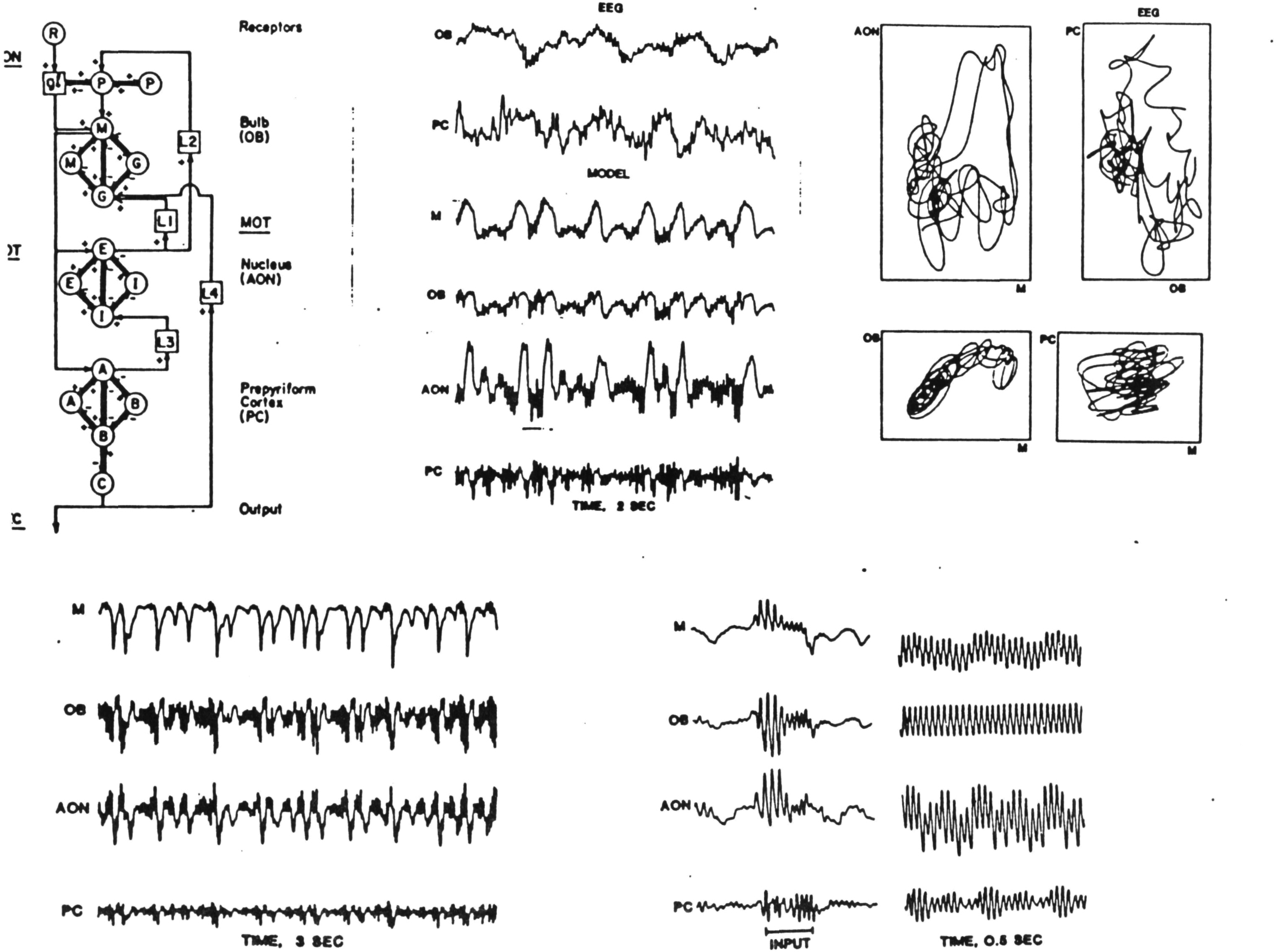




INPUT
1 2 3 4 5 6 7 8
CHANNEL
1
2
3
4
5
6
7
8
A B A B A B O O
OUTPUT



INPUT
1 2 3 4 5 6 7 8
CHANNEL
1
2
3
4
5
6
7
8
A B A B A B O O
OUTPUT



R
P
P
M
M
G
G
L1
L2
E
E
I
I
L3
L4
A
A
B
B
C
Receptors
Bulb
(OB)
MOT
Nucleus
(AON)
Prepyriform
Cortex
(PC)
Output
EEG
OB
PC
MODEL
M
OB
AON
PC
TIME, 2 SEC
AON
M
EEG
PC
OB
OB
M
PC
M
M
OB
AON
PC
TIME, 3 SEC
M
OB
AON
PC
INPUT
TIME, 0.5 SEC

Physiology of the Olfactory Bulb
1967-1987
Walter J. Freeman
Department of Physiology-Anatomy
University of California, Berkeley

A. Physiological Studies

71. Freeman, W.J. Spatial divergence and temporal dispersion in primary olfactory nerve of cat. J. Neurophysiol. 35: 733-761, 1972.

72. Freeman, W.J. Measurement of open-loop responses to electrical stimulation in olfactory bulb of cat. J. Neurophysiol. 35: 745-761, 1972.

73. Freeman, W.J. Measurement of oscillatory responses to electrical stimulation in olfactory bulb of cat. J. Neurophysiol. 35: 762-779, 1972.

74. Freeman, W.J. Depth recording of averaged evoked potential of olfactoory bulb. J. Neurophysiol. 35: 780-796, 1972.

82. Freeman, W.J. Topographic organization of primary olfactory nerve in cat and rabbit as shown by evoked potentials. Electrocephalogr. clin. Neurophysiol. 36: 33-45, 1974.

83. Freeman, W.J. Average transmission distance from mitral tufted to granule cells in olfactory bulb. Electroencephalogr. clin. Neurophysiol. 36: 609-618, 1974.

84. Freeman, W.J. Attenuation of transmission through glomeruli of olfactory bulb on paired shock stimulation. Brain Research 65: 77-90, 1974.

85. Freeman, W.J. Relation of glomerular neuronal activity to glomerular transmission attenuation. Brain Research 65: 91-107, 1974.

86. Freeman, W.J. A model for mutual excitation in a neuron population in olfactory bulb. Trans. IEEE Biomed. Engin. 21: 350-358, 1974.

87. Freeman, W.J. Stability characteristics of positive feedback in a neural population. Trans. IEEE Biomed. Engin. 21: 358-364, 1974.

109. Freeman, W.J. Spatial properties of an EEG event in the olfactory bulb and cortex. Electroenceph. & Clin. Neurophysiol 44: 586-605, 1978.

112. Freeman, W.J. Spatial frequency analysis of an EEG event in the olfactory bulb. In: Multidisciplinary Perspectives in Event-Related Brain potential Research: 531-546. Ed: D.A. Otto. U.S. Govt. Printing Office, EPA-600/9-77-043, 1978.

119. Freeman, W.J. Use of spatial deconvolution to compensate for distortion

of EEG by volume conduction. IEEE Trans. Biomed. Engin. 27: 421-429, 1980c.

120. Bressler, S.L. and Freeman, W.J. Frequency analysis of olfactory system EEG in cat, rabbit and rat. Electroencephalogr. clin. Neurophysiol. 50: 19-24, 1980.

122. Gonzalez-Estrada, M.T. and Freeman, W.J. Effects on carnosine on olfactory bulb EEG, evoked potentials and D.C. potentials. Brain Research 202: 373-386, 1980.

145. Martinez, D.M. and Freeman, W.J. Periglomerular cell action on mitral cell in olfactory bulb shown by current source density analysis. Brain Research, 223-233.

131. Freeman, W.J. and W. Schneider. Changes in spatial patterns of rabbit olfactory EEG with conditioning to odors. Psychophysiology 19: 44-56, 1982.

132. Davis, G.W. and Freeman, W.J. On-line detection of respiratory events applied to behavioral conditioning in rabbits. IEEE Trans. Biomed. Engin. 29: 453-456, 1982.

136. Freeman, W.J., Viana Di Prisco, G. Davis, G.W., Whitney, T.M. Conditioning of relative frequency of sniffing by rabbits to odors. J. Comp. Psychol 97: 12-23, 1983.

151. Viana Di Prisco, G. and Freeman, Odor-related bulbar EEG spatial pattern analysis during appetitive conditioning in rabbits. Behav. Neurosci. 99: pp. 962-978, 1985.

169. Gray, C.M., Freeman, W.J. and Skinner, J.E. Chemical dependencies of learning in the rabbit olfactory bulb: acquisition of the transient spatial-pattern change depends on norepinephrine. Behavioral Neuroscience 100: 585-596, 1986.

153. Freeman, W.J. and Viana Di Prisco, G. Relation of olfactory EEG to behavior: Time series analysis. Behav. Neurosci. 100:753-763, 1986

170. Freeman, W.J. and Baird, B. Relation of olfactory EEG to behavior: Spatial analysis: Behav. Neurosci. 101:393-408, 1987.

179. Freeman, W.J. and Grajski, K.A. Relation of olfactory EEG to behavior: Factor analysis: Behav. Neurosci., 101: 766-777.

B. Theoretical Studies

49. Freeman, W.J. Analysis of function of cerebral cortex by use of control system theory. The Logistics Review 3: 5-40, 1967.

57. Freeman, W.J. Analog simulation of prepyriform cortex in the cat. Math.

BioScio. 2: 181-190, 1968.

65. Freeman, W.J. Spectral analysis of prepyriform averaged evoked potentials in cats. *Journal of Biomed. System 1*: 3-22, 1970.

78. Freeman, W.J. A model of the olfactory system. In: *Neural Modelling, Brain Info. Serv. Report No. 1*. Eds: Brazier, M.A.B., Walter, D.O., and Schneider, E. Univ. Calif., Los Angeles, pp. 41-62, 1973.

90. Ahn, S.M. and Freeman, W.J. Steady-state and limit cycle activity of mass of neurons forming simple feedback loops (II): distributed parameter model. *Kybernetik 16*: 127-132, 1974.

91. Ahn, S.M. and Freeman, W.J. Neural dynamics under noise in the olfactory system. *Bio. Cybernetics 17*:165-168, 1975.

103. Freeman, W.J. and S.M. Ahn. Spatial and temporal characteristic frequencies of interactive neural masses. *Proceedings of IEEE International Conference on Cybernetics and Society*: 279-284, Washington D.C., 1-3 November, 1976.

116. Freeman, W.J. Nonlinear gain mediating cortical stimulus-response relations. *Biol. Cybernetics 35*:237-247, 1979.

117. Freeman, W.J. Nonlinear dynamics of paleocortex manifested in the olfactory EEG. *Biol. Cybernetics 35*: 21-37, 1979.

118. Freeman, W.J. EEG analysis gives model of neuronal template-matching mechanism for sensory search with olfactory bulb. *Biol. Cybernetics. 35*.:221-234, 1979.

119. Freeman, W.J. Use of spatial deconvolution to compensate for distortion of EEG by volume conduction. *IEEE Trans. Biomed. Engin.27*: 421-429, 1980c.

125. Freeman, W.J. A neural mechanism for generalization over equivalent stimuli in the olfactory system. *Amer. Math Soc. SIAM 13*: 25-38,1981.

140. Freeman, W.J. Semi-autonomous control of input by nerve cell assemblies. In: *Proc. IEEE International Conference on Computer Design: VLSI in Computers*. New York, Computer Society Press, 1983, pp21-24.

139. Freeman, W.J. Dynamics of image formation by nerve cell assemblies. in: *Synergetics of the Brain*. Eds. E. Basar, H. Flohr, H. Haken, A.J. Mandell. Berlin, Springer Verlag, 1983, pp 102-121.

172. Freeman, W.J. Stimulation of chaotic EEG patterns with a dynamic model of the olfactory system. *Biological Cybernetics 56*: 139-150, 1987.

173. Freeman, W.J. and Van Dijk, B. Spatial patterns of visual cortical fast EEG during conditioned reflex in a rhesus monkey. *Brain Research*: 422: 267-276, 1987.

C. Reviews

75. Freeman, W.J. Linear analysis of the dynamics of neural masses. Annual Review of Biophysics and Bioengineering 1: 225-226, 1972.

76. Freeman, W.J. Waves, Pulses and the Theory of neural masses. Progress in Theoretical Biology 2: 87-165, 1972

124. Freeman, W.J. A physiological hypothesis on perception. Perspectives Biol. Med. 24: 561-592, 1981.

137. Freeman, W.J. The physiology of mental images. Academic Address. Biol. Psychiatry 18:1107-1125, 1983.

146. Freeman, W.J. La fisiologia de'las imagenes mentales. Salud Mental 7:3-8, 1984.

183. Viana Di Prisco, G. and Freeman, W.J. Representacion espaciotemporal dinamica de la informacion sensorial en el bulbo olfatorio. Acta Cientifica Venezolana 37: 526-531, 1986.

152. Freeman, W.J. and Skarda, C. Nonlinear dynamics, perception, and the EEG; the neo-Sherringtonian view. Brain Res. Reviews 10:147-175, 1985.

154. Freeman, W.J. Petit mal seizure spikes in olfactory bulb and cortex caused by runaway inhibition after exhaustion of excitation. Brain Res. Reviews 11:259-284, 1986.

155. Freeman, W.J. and Viana Di Prisco, G. EEG spatial pattern differences with discriminated odors manifest chaotic and limit cycle attractors in olfactory bulb of rabbits. In: Brain Theory, eds. Palm, G., and Aertsen, A. Berlin, Springer-Verlag., 97-119, 1986.

175. Skarda, C.A. and Freeman, W.J. How brains make chaos to make sense of the world. Brain and Behavioral Science 10: 161-195, 1987.

171. Freeman, W.J. Techniques used in the search for the physiological basis of the EEG. In: Gevins, A., and Remond, A. (eds) Handbook of EEG and Clin. Neurophysiol. Vol 3A, Part 2, Ch. 18. Amsterdam, Elsevier, 1987.

D. Book

96. Freeman, W.J. Mass Action in the Nervous System. 7 Chapters, 489 pp. Academic Press, New York, 1975.

NOTES

Guenter W. Gross, Ph.D.

North Texas State University
Denton, Texas

Dr. Gross received his B.S. in engineering physics from Stevens Institute of Technology in Hoboken, N.J., in 1962. For the following 5 years, he served as a pilot with the U.S. Air Force and participated as project officer in the acceptance testing of the Gemini rendezvous radar system. He returned to graduate school in 1968 and received a Ph.D. in biophysics and neurophysiology from Florida State University in 1973. From 1974 until 1977, he was associated with the Experimental Neuropathology Section at the Max Planck Institute for Psychiatry in Munich. He also worked as a visiting scientist for the Sandoz Corporation in Basel, Switzerland, in the fall of 1977. He is presently an associate professor of neuroscience with the Department of Biological Sciences at North Texas State University (NTSU), and director of the newly established NTSU Center for Networks Neuroscience.

MULTIELECTRODE BURST PATTERN FEATURE EXTRACTION FROM SMALL MAMMALIAN NETWORKS IN CULTURE

Abstract

We are investigating the properties of small (100 to 400 neuron), monolayer networks grown in culture from dissociated mouse spinal cord tissue on glass plates featuring 64 photoetched microelectrodes. These networks form vigorous organotypic activity that becomes organized with time. At 4 weeks after seeding, the cultures exhibit synchronized burst patterns on many electrodes. The spontaneous activity can be maintained for as long as 100 days in culture and is often rhythmic. Network disinhibition via the inhibitory synapse blockers strychnine and bicuculline produces rapid, rhythmic bursting in all cultures with highly stereotyped spike patterns within the bursts, regardless of the nature of the prior spontaneous activity. Data are processed on two levels: (1) burst pattern analysis in terms of burst frequency, duration, and period, and (2) analysis of spike patterns within bursts. Data compression is achieved by burst integration which preserves the character of the spike patterns. Integrated bursts are being classified according to shape and identified with letters, allowing 2 hours of activity on one electrode to be condensed to one page of letter sequences. Pattern recognition and cross-correlations with other electrodes are therewith simplified. In view of the fact that all synapses integrate spike trains, the ignoring of detailed spike information is reasonable and makes real-time, statistical analysis of compressed multichannel data possible.

NOTES

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

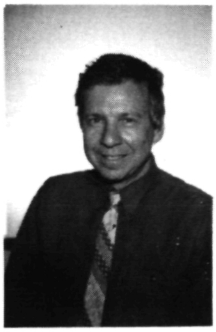
63

S13-51

ABS ONLY

N91-71363

P.1



Michael H. Myers

TRW MEAD
San Diego, California

Mr. Myers is a graduate of the Massachusetts Institute of Technology, and of Purdue University and is the manager of Neural Network Programs at TRW MEAD Rancho Carmel Artificial Neural System Center. He is a communications systems engineer with both digital and analog processing experience in military communications, digital avionics, integrated aircraft antenna systems, radar, and sixth-generation computing (artificial neural systems). He has had international marketing experience with both General Electric Information Services and the (former) Singer Business Machines Division, and speaks several foreign languages including Russian, German, Swedish, and French. In the past 5 years, he has received six patents for signal processing inventions in the areas of waveform demodulation, synchronization, and adaptive interference cancellation, and has presented numerous technical papers. Mr. Myers organized and set up the first computer time-sharing link between the United States and the U.S.S.R. He was appointed by the Assistant Secretary of Commerce to serve a term on the Computer Systems Technical Advisory Committee of the Office of Export Administration, having a charter to develop U.S. export licensing policy guidelines for the sale and/or transfer of advanced technology to foreign entities, including the Eastern bloc.

A HYBRID CONNECTIONIST/AI ARCHITECTURE FOR REFLEXIVE, REFLECTIVE, AND EXPLORATORY SYSTEMS

Abstract

Implementing autonomous systems capable of operation in both familiar and unfamiliar environments is a goal being hotly pursued in many communities, both military and commercial. This presentation discusses an autonomous system philosophy and architecture which combines both trainable and non-trainable artificial neural network elements for both mapping and dynamic system simulation, with rule-based decision-directed structures. The architecture presented should manifest interesting behavior characteristics, including the ability to handle an unbounded dimensional environment, the ability to provide rapid reflex response as well to enable reflecting and planning actions, and the ability to execute generic exploratory behaviors designed to enhance knowledge of the environment when it is ambiguous. A methodology for embedding both discrete rules and apprentice learning in the system is used to initialize the autonomous system. It is hoped that this presentation will help air some of the issues faced by neural network designers attempting to merge connectionist and rule-based technologies.

NOTES

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

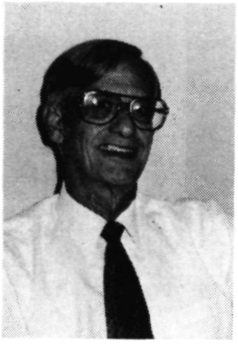
64

S14-62

15314

N91-71364

p.16



William Siler, Ph.D.

Mote Marine Science
Laboratory
Sarasota, Florida

Dr. Siler received his M.S. degree in fluid dynamics from Stevens Institute of Technology and his Ph.D. in biology from City University of New York. He is a senior research associate at the Mote Marine Laboratory in Sarasota, Florida (since 1987). From 1959 to 1965, Dr. Siler was supervisor of the computer facility and radiological physics section at Memorial Sloan-Kettering Cancer Center in Birmingham, Alabama. He was an associate professor and chairman of the Biomedical Computer Science Program at Downstate Medical Center from 1965 to 1972; professor and chairman of the biomathematics department at the University of Alabama at Birmingham from 1972 to 1980; and director of clinical computing at Carraway Methodist Medical Center in Birmingham from 1980 to 1987. From 1970 to 1974, Dr. Siler was a member of the computer and biomathematical sciences study section of the National Institute of Health. He was Data Processing Professional of the Year of the Data Processing Managers' Association in 1981, and a member of the Biostatistics and Epidemiology Contract Review Committee of the National Cancer Institute from 1981 to 1985 - serving the committee as chairman from 1984 to 1985.

FLOPS: A PARALLEL-RULE-FIRING FUZZY EXPERT SYSTEM SHELL

Abstract

The use of fuzzy systems theory as a basis for expert systems is reviewed with particular reference to a fuzzy expert system having rules that are fired in parallel. Examples are given of fuzzy sets, fuzzy numbers, and fuzzy logic, and their use in pattern recognition and process control. Fuzzy systems theory may be looked upon as furnishing ways of processing information which is uncertain, imprecise, vague, ambiguous, or contradictory. This paper is concentrated on the use of fuzzy systems theory in processing information which is ambiguous or contradictory, rather than uncertain, vague, or imprecise. We also show how advantages can be reaped from the intentional introduction of ambiguities in description, even in a field as objective as process control.

FLOPS - A GENERAL-PURPOSE

FUZZY EXPERT SYSTEM SHELL

William Siler, Ph.D.

Kemp-Carraway Heart Institute, Birmingham, AL 35234
and
Mote Marine Laboratory, Sarasota, FL 34236, USA

April, 1988

PROCEDURAL LANGUAGES (FORTRAN, PASCAL C)

and

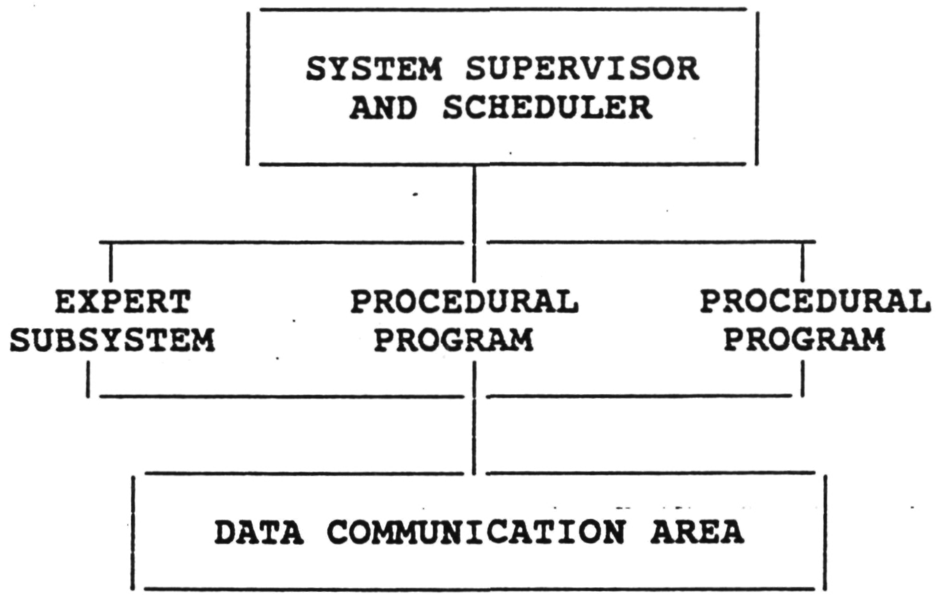
NON-PROCEDURAL LANGUAGES (EXPERT PRODUCTION SYSTEMS)

Advantages and Disadvantages for Problem Solving:

	PROCEDURAL	NON-PROCEDURAL
Speed	Fast	Very Slow
Numerical Computation	Very Good	Very Bad
Symbolic Computation (Reasoning)	Poor	Excellent

Best power obtained from a mix of procedural and non-procedural languages

BLACKBOARD EXPERT SYSTEM FRAMEWORK



BLACKBOARD SYSTEM PERMITS COMBINING ADVANTAGES OF BOTH PROCEDURAL AND NON-PROCEDURAL LANGUAGES.

REQUIREMENTS:

Expert system should be able to call programs or systems in any other language; control should revert to expert system on program or system termination.

A flexible common framework should be chosen for data, so the programs can communicate effectively and conveniently with each other.

Few expert system shells meet these requirements.

HANDLING UNCERTAINTIES

Candidate Methods:

<u>Ad hoc methods</u> (e.g. Mycin, M1)

<u>Bayes' Theorem</u> (e.g. Prospector derivatives)

<u>Dempster-Shafer Theory</u>

<u>Fuzzy Systems Theory</u>

HANDLING AMBIGUITIES:

An ambiguity: the situation when more than one of several possibilities might be true.

A Contradiction: only one of several possibilities is true, but we don't know which one.

Candidate Methods:

<u>Ad hoc methods</u> (virtually non-existent)

<u>Probability distributions</u> (theoretically possible, but very awkward and seldom if ever done)

<u>Fuzzy Systems Theory</u> (extremely easy)

EXAMPLES OF A FUZZY SET:

ATHLETES IN MASH 4077

Member	*Grade of Membership*
Lt. Col. Penobscot	1.0
Father Mulcahey	0.8
Major Houlihan	0.6
B. J. Honeycutt	0.5
Klinger	0.4
Hawkeye Pierce	0.15

OBJECTS ON A RADAR SCREEN

Member	*Grade of Membership*
Commercial Airliner	0.02
Military Jet	0.01
Light Plane	0.13
Powered Ultralight	0.44
Hang Glider	0.63
Santa Claus	0.99

EXAMPLE OF A FUZZY NUMBER: A FUZZY TWO

```
Grade of
Membership

1.0 +                  *
    -                 * *
    -                *   *
    -               *     *
    -              *       *
0.5 +             *         *
    -            *           *
    -           *             *
    -          *               *
    -         *                 *
0.0 ***********--------+---------************
    0        1         2         3          4

         Member
```

EXAMPLE OF FUZZY LOGIC:

My talk will be a success if the material is interesting, the visual material good and the audience is really interested or if the talk is given by a very exciting speaker.

```
Rule 1: ( ( Material = interesting
                  AND
            visuals = good
                  AND
            audience = interested )
                  OR
          ( speaker = exciting )
                  -->
          talk = success
```

Confidence that (material = interesting) = 0.75
Confidence that (visuals = good) = 0.6
Confidence that (audience = interested) = 0.88
Confidence that (speaker = exciting) = 0.33

Combined confidence:
  first clause, min(0.75, 0.6, 0.88) = 0.6
  second clause = 0.33
  first OR second clause, max(0.33, 0.60) = 0.60

(AND Rule: A chain is no stronger than its weakest link.)

APPLICATION: PATTERN RECOGNITION

General Scheme:

1. Feature extraction by procedural language programs.

2. Assign word descriptors to features using fuzzy sets to handle ambiguities.

3. Assign preliminary classifications using fuzzy sets to handle contradictions.

4. Resolve contradictory preliminary classifications to obtain final classifications. Usually means pulling in additional information as required.

ASSIGNING WORD DESCRIPTORS TO NUMERIC FEATURES:

AREA of image region to be classified = a numeric feature

SIZE of image region is a fuzzy set of word descriptors:

SIZE = {TEENY SMALL MEDIUM LARGE HUGE}

RULE (in English): if for any region the AREA is approximately less than or equal to a fuzzy 100 plus or minus 50 then the SIZE is TEENY.

In FLOPS:

```
rule ( region ^area ~<= 100,50,0 )
      --> modify 1 ^size.TEENY ;
```

Other descriptors:

xbar = numeric feature,
xpos = {FAR_LEFT LEFT CENTER RIGHT FAR_RIGHT} = fuzzy set

ybar = numeric feature,
ypos = {HIGHEST HIGH MIDDLE LOW LOWEST} =fuzzy set

CLASSIFICATION RULES:

Classification fuzzy set used in echocardiogram classification:

class = {LV RV LA RA LV+LA RV+RA ARTIFACT PAPILLARY ...}

RULE (in English) If in any region the size is SMALL and the x-position is CENTER and the y-position is HIGHEST then it is likely to be an ARTIFACT.

In FLOPS:

```
rule ( region ^size.SMALL ^xpos.CENTER
              ^ypos.HIGHEST )
     -->
     modify 1 ^class.ARTIFACT ;
```

MATCHING OBSERVED PATTERN AGAINST LIBRARY PATTERN

For illustration, we match only one fuzzy set, that for **size**. In general, more than one fuzzy set would be simultaneously matched.

Fuzzy Set Size:

Member	Observed Grade-Of-Membership	Pattern_1 Grade-Of-Membership	Observed AND Pattern_1 Grade-Of-Membership
Very Large	0	0.15	0
Large	0.04	1.00	0.04
Medium	1.00	0.45	0.45
Small	0.65	0	0
Very Small	0	0	0

Confidence in match = match on Very Large OR Large OR ....
= max(0, 0.04, 0.45, 0, 0)
= 0.45
= grade of membership of PATTERN_1 in fuzzy set of classifications.

(We are on our way toward a simple and reliable pattern-matching technique, making use of the ambiguities in the word descriptor fuzzy sets.)

FUZZY PROCESS CONTROL

Especially useful when we do not have a mathematical model of the process.

Block Diagram:

```
                          Set Point
                             |
                             |
                            _v_
 ___________               |   |               ____     _____________
|           |   Output     |   |    Error     |    |   |             |
|  Process  |------------->| - |------------->| A/D |-->| Computer    |
|___________|              |___|              |____|   |             |
      ^                                        ____    | Control     |
      |        Controller Output              |    |   | Program     |
      ----------------------------------------| D/A |<--|             |
                                              |____|   |_____________|
```

SIMPLE PROCESS CONTROL PROGRAM:

1. Convert error, rate of change of error to fuzzy set of word descriptors such as NEGATIVE_SMALL, POSITIVE_LARGE (fuzzification).

2. Use control rules to obtain fuzzy set of word descriptors for controller output.

3. Convert fuzzy set for controller output to voltage (defuzzification).

Typical control rules:

IF error is POSITIVE_SMALL and rate is ZERO then controller-output is NEGATIVE_SMALL.

In FLOPS:

```
rule ( process ^error.P_SMALL ^rate.ZERO )
     ( controller )
     -->
     modify 2 ^output.N_SMALL ;
```

FLOPS: A FUZZY EXPERT SYSTEM SHELL.

Features:

(1) Deductive reasoning is emulated by a conventional sequential-rule-firing mode; inductive reasoning is emulated by a unique parallel-rule-firing mode which in turn emulates a non-von-Neumann parallel computer.

(2) Data types include integers; floats; strings; fuzzy numbers; fuzzy sets and confidence levels.

(3) Two external file types are provided: Type I, FLOPS programs and commands; and Type II, "flat file" relational data base format.

(4) External programs written in any language may be called in the same manner as a DOS call: program name plus command string. With (3), provides a blackboard system.

(5) A basic truth maintenance system is provided based on monotonic fuzzy logic; this may be overridden by the programmer to provide fully non-monotonic logic.

(6) Backtracking is fully automatic in sequential mode. Since in parallel mode all fireable rules are fired concurrently, backtracking is irrelevant.

SUMMARY

(1) Fuzzy systems theory permits handling uncertainties, ambiguities and contradictions in a mathematically convenient and rigorous fashion. It may be used both in procedural and non-procedural languages. When employed in an expert system, a system shell should be written or selected which incorporates these basic features:

Confidence factors for strings, floats and integers;
Discrete fuzzy sets;
Fuzzy numbers;
Approximate numerical comparison operators.

(2) Although expert production systems are too slow to permit their unassisted use in most online applications, they may be used in conjunction with procedural language programs in a blackboard system to combine the reasoning skills of an expert system with the computational ability of procedural language programs.

(3) While fuzzy techniques are very powerful, they are unfamiliar to most American engineers and scientists. Study and practice in their use is required.

NOTES

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

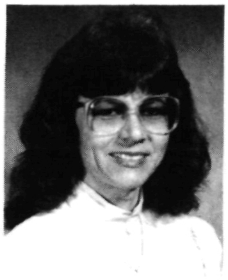
65

S15-63

15315

P.33

N91-71365



Maria Zemankova, Ph.D.

University of Tennessee
Knoxville, Tennessee

Dr. Zemankova received her B.S. degree from the American University in Cairo in 1977, majoring in mathematics and computer science with a minor in psychology. Her M.S. and Ph.D. degrees were in computer science from Florida State University in 1979 and 1983, respectively. In 1983, she worked on the design and implementation of a relational data base "team-up" at Unlimited Processing, Inc., in Jacksonville, Florida. Since 1984, Dr. Zemankova has been an assistant professor in the Department of Computer Science at the University of Tennessee, Knoxville, and is also engaged in research collaboration with the Oak Ridge National Laboratory. She spent January to August 1987 as a visiting research assistant professor in the Department of Computer Science, University of Illinois, Urbana. Her research interests include intelligent information systems, machine learning, reasoning under uncertainty, and knowledge representation. Dr. Zemankova has authored a book entitled "Fuzzy Relational Data Bases - a Key to Expert Systems" (1984), and several papers on fuzzy intelligent information systems, and on applications of fuzzy set and rough set theories to information systems design and machine learning. Her professional affiliations include AAI, ACM, IEEE, IFSA, NAFIPS, and Sigma xi, and she is an associate editor of the International Journal of Approximate Reasoning.

INTELLIGENT INFORMATION SYSTEMS WITH LEARNING CAPABILITIES

Abstract

An intelligent information system is designed to derive information (that may not be explicitly stored in the data base) by application of rules for inferring plausible answers to queries. The system is divided into the knowledge base (KB) and the inference engine (IE). The KB can be further partitioned into a factual base (FB) and an explanatory base (EB). The FB is used for storing facts (data) that may be imprecise or incomplete, and the EB contains knowledge; i.e., flexible (fuzzy) concepts, relationships, or rules that are used to interpret the available data. The IE is designed to perform flexible reasoning. Clearly, the "intelligence" of the system depends on the knowledge available in the KB and the types of inferences that the IE is capable of performing. An experimental system (APPLAUSE) is discussed, together with demonstration of system function in the knowledge acquisition and querying modes, including its explanatory capabilities.

INTELLIGENT INFORMATION SYSTEMS

WITH LEARNING CAPABILITIES

Maria Zemankova

Department of Computer Science
University of Tennessee
Knoxville, Tennessee

THEME

- Make Machines Behave Intelligently

Marks of intelligence:

- Learning Capability
- Reasoning with Insufficient, Unreliable or Imprecise Data
- Reasoning Under Resource Constraints
- Creativity- Discovery

PLAUSIBLE REASONING

- A Core theory- proposed by Collins and Michalski
- Modifications and Implementation

MODEL

- Hierarchical Organization of Knowledge
- Mechanism to Manipulate Incomplete and Uncertain Knowledge Base
- Domain Independent Inference Mechanism
- Theory is Operationalized with Chemical Periodic Table as a Test Domain

KNOWLEDGE REPRESENTATION

ELEMENTS OF EXPRESSIONS

- Arguments
- Descriptors
- References
- Terms
- Facts - veracity-μ, frequency-ϕ, confidence-γ
- Dependency - forward-backward dependencies
- Implication - forward-backward implications
- Hierarchies - generalization, specialization
- Similarity - context, dominance, typicality

ELEMENTS OF EXPRESSIONS:

<u>Descriptor:</u> d.

breed	attribute
temperature	attribute, function
flies	predicate

<u>Terms:</u> $d_1(a_1)$, or $d_2(a_1, a_2, \ldots)$

$breed(Fido)$
$temperature(latitude,\ altitude)$
$temperature(place)$

<u>References:</u> $r_1, \{r_1, ...\}$.

4	integer
true	logical
*group*6	hierarchical

Factual statements: $d_1(a_1) = r_1 : [\mu, \gamma_\mu, \phi, \gamma_\phi]$

- μ- veracity: Veracity indicates degree with which reference r_1 is applicable to descriptor-argument pair.
- ϕ- frequency: Frequency indicates proportion of argument for which the reference is a valid description of the descriptor-argument pair.
- γ_μ, γ_ϕ- confidences in μ, ϕ.

Examples:

$$density(aluminum) = 2.7 : [1, .99, 1, 1]$$
$$is\_old(john) = yes : [.7, .9, 1, 1]$$
$$engine\_type(car) = 4\_cylinder : [1, 1, .8, .95]$$

Dependency between terms:

$$d_1(a_1) \longleftrightarrow d_2(a_2) : [\alpha, \gamma_\alpha, \beta, \gamma_\beta]$$

- α, β- forward and backward dependency strengths

$$is\_philosopher(X) \longleftrightarrow is\_greek(X) : [.5, .8, .0001, .8]$$

Implications between factual statements:

$$d_1(a_1) = r_1 \Longleftrightarrow d_2(a_2) = r_2 : [\alpha, \gamma_\alpha, \beta, \gamma_\beta]$$

$$grain(place) = rice \Longleftrightarrow$$
$$rain(place) = [80..120in] : [.9, .9, .5, .8]$$

The implications can also be encoded by functions

$$d_1(a_1) = r_1 \Longleftrightarrow d_2(a_2) = f(r_1).$$

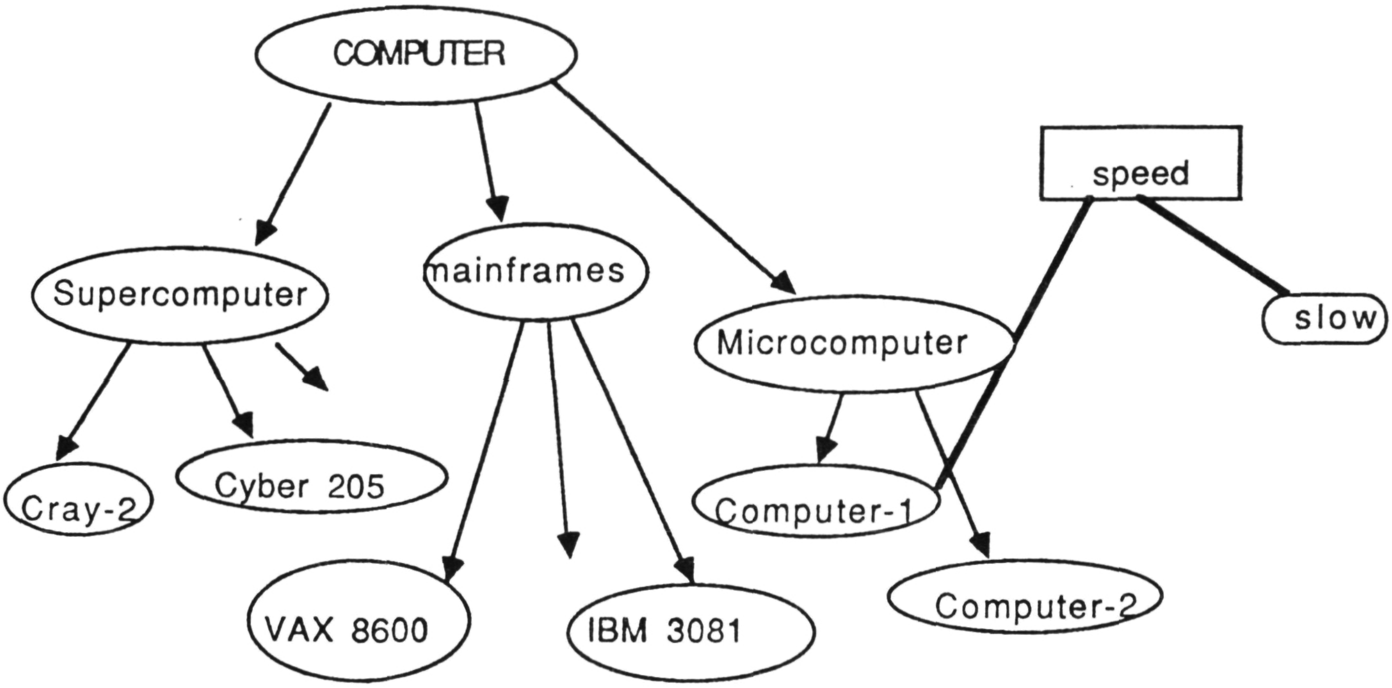
$$radius(circle) = r \Longleftrightarrow$$
$$area(inscribed\ square) = 2r^2 : [1, 1, 1, 1]$$

TRANSFORMATIONS

The transforms A GEN, A SPEC, A SIM, R GEN, R SPEC, R SIM allow traversal in a hierarchy, in the process of inference. Simplified (no parameters) applications of the transforms are given below.

- A GEN

$$\frac{\begin{array}{l}\text{speed}(\text{computer\_1}) = \text{slow} \\ \text{micro\_computer} = \text{gen}(\text{computer\_1}):\ \text{cx} = \text{alu\_size} \\ \text{alu\_size}(\text{COMPUTER}) \longleftrightarrow \text{speed}(\text{COMPUTER}) \\ \text{micro\_computer} = \text{spec}(\text{COMPUTER})\end{array}}{\text{speed}(\text{micro\_computer}) = \text{slow}}$$



A GEN Transformation

- A SPEC

$$\frac{\begin{array}{l}\text{height(basketball\_player)} = \text{tall}\\ \text{karim} = \text{spec(basketball\_player)}\end{array}}{\text{height(karim)} = \text{tall}}$$

- A SIM

$$\frac{\begin{array}{l}\text{economy(singapore)} = \text{strong}\\ \text{hongkong} = \text{sim(singapore): cx} = \text{economic structure}\end{array}}{\text{economy(hongkong)} = \text{strong}}$$

- R GEN

$$\frac{\begin{array}{l}\text{reacts\_with}(\text{potassium}) = \text{chlorine}\\ \text{group7} = \text{gen}(\text{chlorine})\end{array}}{\text{reacts\_with}(\text{potassium}) = \text{group7}}$$

- R SPEC

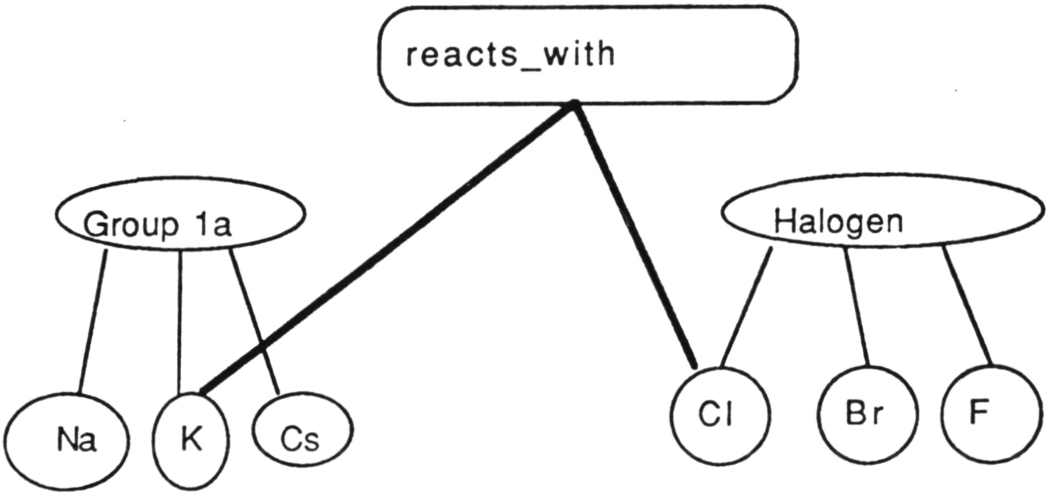
$$\frac{\begin{array}{l}\mathsf{likes(mary)} = \mathsf{carbonated\_drinks} \\ \mathsf{coke} = \mathsf{spec(carbonated\_drink)}\end{array}}{\mathsf{likes(mary)} = \mathsf{coke}}$$

- R SIM

$$\frac{\begin{array}{c}\text{habitat(whales)} = \text{atlantic\_ocean} \\ \text{pacific\_ocean} = \text{sim(atlantic\_ocean)}\end{array}}{\text{habitat(whales)} = \text{pacific\_ocean}}$$

- Combination of A SIM and R GEN

$$\frac{\dfrac{\begin{array}{l}\text{reacts\_with(potassium)} = \text{chlorine} \\ \text{sodium} = \text{sim(potassium)}\end{array}}{\text{reacts\_with(sodium)} = \text{chlorine}}\ \text{A SIM} \qquad \text{group7} = \text{gen(chlorine)}}{\text{reacts\_with(sodium)} = \text{group7}}\ \text{R GEN}$$



A SIM, R GEN Combination

Theory of Plausible Reasoning and its Implementation.

Collins and Michalski introduced a theory to model human plausible reasoning. APPLAUSE is an implementation of an extended and modified version of the theory. The methodology is eminently suitable to reason in the domains where knowledge is organized hierarchically. The theory provides mechanisms to manipulate the knowledge base in case of incomplete and uncertain knowledge. Some features of the theory are highlighted with examples from chemical periodic table.

QUERIES

Form:

$$descriptor(argument) = ref?/[\mu, \gamma_\mu, \phi, \gamma_\phi]? \qquad (1)$$

Aim:

In query (1) the system is to retrieve best reference value together with the estimated parameters. The best reference is one with highest $\mu * \gamma_\mu * \phi * \gamma_\phi$ product.

- Type checking is performed for arguments and descriptors and references when applicable.

ALGORITHM for processing Queries:

- get_query(Q)
- if (get_fact(Q) successful) then – report retrieved information, **exit.**
- elseif reasoning_depth_counter $>$ depth_limit then
  – combine whatever evidence available and exit.
- else
  - increment depth_counter by one.
  - Dep := set of dependencies/implications, such that descriptor occurs in RHS and $\alpha * \gamma_\alpha >$ threshold T.
  - sort dependencies and implications according to decreasing $\alpha * \gamma_\alpha$ (gather strongest evidence first).
  - repeat until no more dependencies.
    * evaluate LHS of dependency or implication. If necessary call this routine to evaluate LHS.
    * apply suitable transforms such as A GEN, A SPEC, A SIM and compute RHS, decrement depth_counter by one and exit.
- combine evidences:
  - choose best γ_μ, γ_ϕ or $\mu * \gamma_\mu, \phi * \gamma_\phi$ products, for type 1 or type 2 query respectively.
  - compute union and intersection of ranges to give upper and lower bounds on the range of the conclusion respectively.

EXAMPLE

Given:

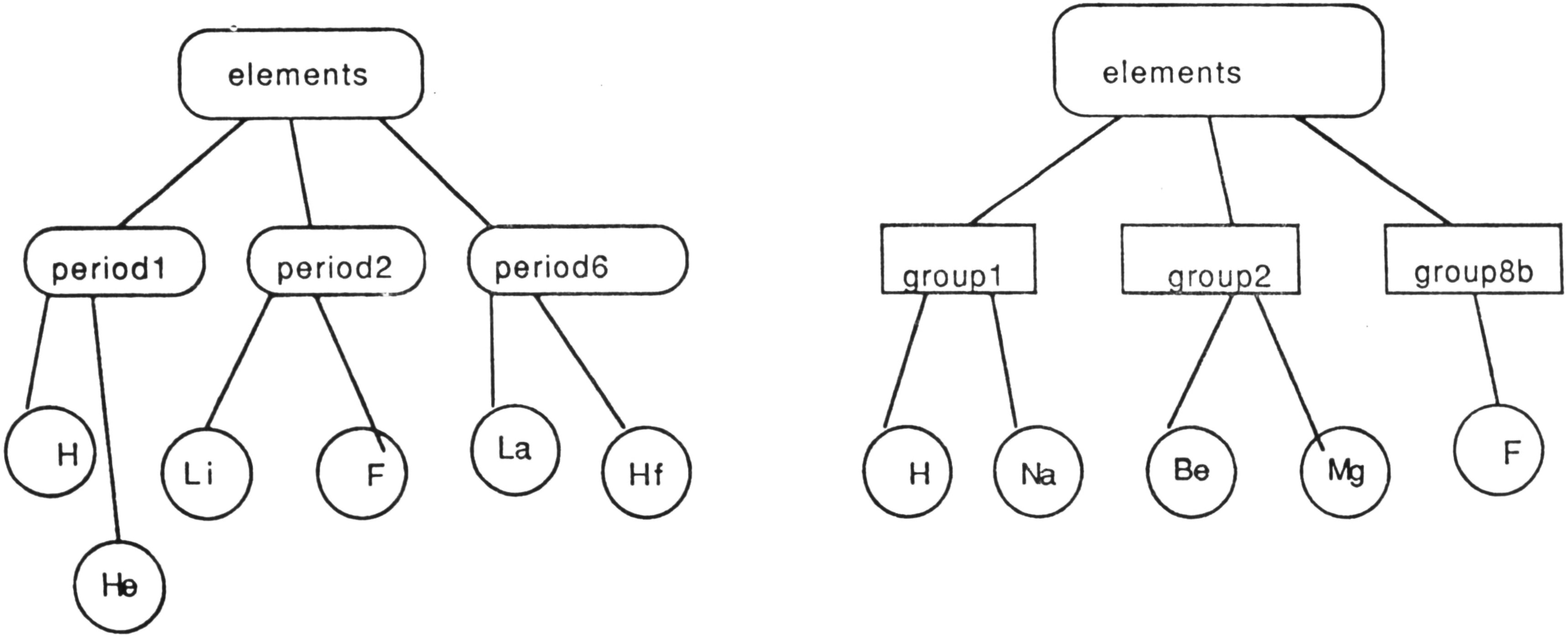
- Group8a consists of gases [He, Ne, Ar, Kr, Xe, Rn].
- Boiling points of only 4 gases are known.

  [He/-269, Ne/-246, Ar/-185, Xe/-108].

Query: Find boiling point of Kr.

Process:

- Statistical analysis will be made to see if it is reasonable to aggregate the boiling points into a range and propagate it to a parent node.
- Kr has two parents, group8a and period4.
- Suitability of generalization is tested in both hierarchies.
- The better one is selected for inference.



Two possible hierarchies for periodic table

Criteria for generalization:

- Low standard deviation of residuals
- Generalize from large number of points.
- Low range of residual errors (fewer outliers)
- Presence of functional dependencies with characteristics similar to those in the neighboring classes.
- 'Causal connection'

Criteria for generalization:

- Low standard deviation of residuals
- Generalize from large number of points.
- Low range of residual errors (fewer outliers)
- Presence of functional dependencies with characteristics similar to those in the neighboring classes.
- 'Causal connection'

Total # of Points = 100
Total # of Variables = 3
Names of the attributes GROUP PERIOD ATNUM
A suitable hierarchy is to be decided for attribute ATNUM

Evaluating Group as the Primary Attribute for Classification
Total Number of Distinct Classes = 18

Class #	Group #	Group	Attr Max	Attr Min	Attr Avg	Std dev of residuals as % of AttrAvg
1	1	IA	87	1	30.43	35.9
2	2	IIA	88	4	36.33	24.6
3	2.1	IIIB	100	21	74.27	6.3
4	2.2	IVB	72	22	44.66	12.8
5	2.3	VB	73	23	45.66	12.2
6	2.4	VIB	74	24	46.66	12.2
7	2.5	VIIB	75	25	47.66	12.0
8	2.6	VIII	76	26	48.66	11.7
9	2.65	VIII	77	27	49.66	11.5
10	2.7	VIII	78	28	50.66	11.3
11	2.8	IB	79	29	51.66	11.0
12	2.9	IIB	80	30	52.66	10.8
13	3	IIIA	81	5	35.8	21.3
14	4	IVA	82	6	36.8	20.7
15	5	VA	83	7	37.8	20.2
16	6	VIA	84	8	38.8	19.6
17	7	VIIA	85	9	39.8	19.1
18	8	VIIIA	86	2	34.33	26.1

Evaluating Period as the Primary Attribute for Classification
Total Number of Distinct Classes = 7

Class#	Period#	AttrMax	AttrMin	AttrAvg	Std dev of residuals as % of AttrAvg
1	1	2	1	1.5	26.1
2	2	10	3	6.5	0.0
3	3	18	11	14.5	0.0
4	4	36	19	27.5	2.7
5	5	54	37	45.5	2.7
6	6	86	55	70.5	6.5
7	7	100	87	93.5	3.8

Generalization	group8a	period4
BP range	[-108 .. -269]	[58 .. 3450]
slope m	42	-416
std dev of residuals σ_r	5%	46%
% intersection with neighboring class x	60%	100%
number of points n	4	17
Computed α	0.88	0.8
Computed γ_α	0.93	.3

The equation (implication in a functional form) derived by best line fit method:

BP = -317 + 41.8*period (valid for group8a).

The parameters α and γ_α are estimated by evaluating compliance to the criteria for generalizing.

$$\alpha = (1 - 0.2 * x)$$

$$\gamma_\alpha = 0.5 * \left(\frac{m_{max} - m}{m_{max} - m_{min}}\right) + 0.4 * (1 - \sigma_r) + 0.1 * \left(\frac{n}{n + 4}\right)$$

Tabulated factors favor generalization in group8a rather than in period 4.

PARAMETERS FOR THE DERIVED CONCLUSION:

Derivation using A SPEC without functional dependency

$$BP(Kr) = [-108, -269]$$

Parameters $[\mu, \gamma_\mu, \phi, \gamma_\phi]$ are directly inherited from the parent node, however the precision of the answer is low.

The answer is made more precise (narrower range) by using functional dependencies discovered in the related elements.

Derivation using A SPEC together with functional dependency.

Assume τ and γ_τ for Kr within group8a = .9, .95

These can be estimated by evaluating common relevant features among the siblings.

BP(Kr) = -317 + 41.8*4 = -149.8

$$\begin{aligned}
\mu_c &= \mu_1 = 1 \\
\gamma_{\mu_c} &= \gamma_\mu * \alpha * \gamma_\alpha * \tau * \gamma_\tau \\
&= 1 * .9 * .9 * .9 * .95 = 0.6925 \\
\phi_c &= \phi_1 = 1 \\
\gamma_{\phi_c} &= \gamma_{\phi_1} * \alpha * \gamma_\alpha * \tau * \gamma_\tau \\
&= 1 * .9 * .9 * .9 * .95 = 0.6925
\end{aligned}$$

Derivation using A SIM transform.

Find elements similar to Kr in some context which affects boiling point.

Suppose, relevant context is

CX = (.7*group + .3*period) Rule 1

and the dependency is given by,

CX -> boiling point: $\alpha = .75, \gamma_\alpha = 1$; Rule 2

- α and γ_α estimated by global multiple regression analysis.
- Localize the search space within the neighborhood of the argument in question
- Similarity σ and γ_σ are computed according to the formulas:

$$\sigma(Arg_1, Arg_2) = \sum W_i * \sigma(attr_i(Arg_1), attr_i(Arg_2))$$

$$\gamma_\sigma(Arg_1, Arg_2) = \sum W_i * \gamma_\sigma(attr_i(Arg_1), attr_i(Arg_2))$$

where, the weights W_i are normalized such that the sum of weights is 1.

Assuming pairwise similarity σ and γ_σ values

$$\sigma(\text{gr8a, gr7a}) = .2; \quad \gamma_\sigma = .95$$
$$\sigma(\text{per4, per3}) = .8; \quad \gamma_\sigma = .95$$
$$\sigma(\text{per4, per5}) = .7; \quad \gamma_\sigma = .95$$

and W_i given by context in Rule 1, we get

Elem	Gr.	Per.	σ(Kr, Elem)	γ_σ(Kr, Elem)
Ar	8a	3	$.7*1 + .3*.8 = .94$	.95
Xe	8a	5	$.7*1 + .3*.7 = .91$	.95
Cl	7a	3	$.7*.2 + .3*.8 = .38$	.95
Br	7a	4	$.7*.2 + .3*1 = .44$	.95
I	7a	5	$.7*.2 + .3*.7 = .35$	.95

Disregard elements with $\sigma * \gamma_\sigma <$ threshold T.

Similarity transform reference and parameter estimation:

$$\text{BP(Kr)} = \text{BP(Element)}\ [\mu, \gamma_\mu, \phi, \gamma_\phi]$$

Ar: $\text{BP(Ar)} = \text{-185}\ [\mu = 1, \gamma_\mu = 1, \phi = 1, \gamma_\phi = 1]$

$$\begin{aligned}
&\text{BP(Kr)} = \text{BP(Ar)} = \text{-185.94} \\
&\mu_c = \mu = 1, \\
&\gamma_{\mu_c} = \gamma_\mu * \sigma * \gamma_\sigma * \alpha * \gamma_\alpha \\
&\quad = 1 * .94 * .95 * .7 * 1 = .625 \\
&\phi_c = \phi = 1 \\
&\gamma_{\phi_c} = \gamma_\phi * \sigma * \gamma_\sigma * \alpha * \gamma_\alpha
\end{aligned}$$

$$1 * .94 * .95 * .7 * 1 = .625$$

similarly,

Xe: $\text{BP(Xe)} = \text{-108}\ [1, 1, 1, 1]$

$$\begin{aligned}
&\text{BP(Kr)} = \text{BP(Xe)} = \text{-108.91} \\
&\mu = 1 \\
&\gamma_\mu = 1 * .91 * .95 * .7 * 1 = .605 \\
&\phi_c = 1 \\
&\gamma_{\phi_c} = 1 * .91 * .95 * .7 * 1 = .605
\end{aligned}$$

COMBINATION OF EVIDENCES

- Take the reference value of the result as the weighted average value of the BPs where the weights are decided by $\sigma * \gamma_\sigma$ product.

- The parameters are taken as the weighted average of the evidences.

BP(Kr)= [(BP(Ar)$*\sigma$(Kr,Ar) $+$ BP(Xe)$*\sigma$(Kr,Xe)]/$(\Sigma\, \sigma_i)$

BP(Kr) $= (-185 * .94 + -108 * .91)/(.94 + .91) = -147.1$

$\mu_c = \Sigma\, \mu_i / N$ (if $\Delta\mu$ not large)

$(1 + 1)/2 = 1$

$$\gamma_{\mu_c} = \Sigma\, \gamma_{\mu_i} * \sigma * \alpha * \gamma_\alpha / N$$
$$= (1 * .94 * .75 * 1 + 1 * .91 * .75 * 1)/(1 + 1) = .6175$$

$\phi_c = \Sigma\, \phi_i / N$

$(1 + 1)/2 = 1$

$$\gamma_{\phi_c} = \Sigma\, \gamma_{\phi_i} * \sigma * \alpha * \gamma_\alpha / N$$
$$= (1 * .94 * .75 * 1 + 1 * .91 * .75 * 1) = .6175$$

Comparison of parameters:

Parameter	A SPEC	A SIM	Actual
BP	-149.8	-147	-152
μ	1	1	-
γ_μ	.6925	.6175	-
ϕ	1	1	-
γ_ϕ	.6925	.6175	-

Choose results obtained by A SPEC since it yields inference with higher confidence.

CONCLUSION

Plausible Reasoning provides a useful mechanism to manipulate available knowledge base to infer conclusions not arrivable by traditional logic.

NOTES

S16-63
ABS. ONLY

N91-71366 153/6

Pentti Kanerva

RIACS
NASA Ames Research
Center

UNDERSTANDING INFORMATION PROCESSING IN ANIMALS AS A WAY TO BUILDING INTELLIGENT ROBOTS

Abstract

One motive for the study of artificial neural systems is that knowledge derived from understanding the brain (and its billions of neurons) can be exploited and utilized in the building of machines. We hope to build machines with observational, adaptive, and manipulative powers resembling those of animals and humans. The machines could be used to reduce human exposure to conditions hostile to humans. Severe, dangerous, and unknown environments; long missions to remote locations; and critical, but tedious tasks represent such conditions. Fundamental understanding of nervous systems may be our only hope for building such machines, as no machine built on other principles has come close to achieving this goal.

NOTES

67

510-63

15317

N91-71367 P. 29

Claude A. Cruz

Plexus Systems
Nashua, New Hampshire

Claude Cruz received his B.S. in electrical and biomedical engineering from the University of Southern California, and a joint M.S. in these same subjects from the University of Illinois. Following graduate school, Mr. Cruz spent 11 years with IBM. The first 5 years of this were devoted to various circuit and systems design functions. This was followed by a transfer to the IBM Palo Alto, California, Scientific Center, where Mr. Cruz's initial work involved processor architecture and artificial intelligence (particularly expert systems technology). In 1981, Mr. Cruz founded a wide-ranging IBM neural-net project. This work covered neural-net theory, applications, and implementation techniques. In particular, it led to the creation of three generations of sophisticated tools for neural-net experimentation. These include a novel network "compiler", an interactive network debug environment, and a parallel digital network emulator. In May of 1987, Mr. Cruz left IBM to become a neural-net consultant, and to pave the way for founding a neural-net company (Plexus Systems).

KNOWLEDGE PROCESSING USING NEURAL NETWORKS

Abstract

Artificial neural networks (ANN's) are novel information processing mechanisms. Such networks are based on formal models of the function and organization of biological nerve nets. Neural nets possess several traits which make them an attractive substrate for artificial intelligence applications. They are an inherently parallel processing mechanism promising great speed of operation. In addition, most artificial neural nets are adaptive in that the behavior of the network may change over time as a function of its "experience" (cumulative operating history). Traditional AI involves the emulation of very high-level behaviors and cognitive mechanisms. Neural nets are a comparatively simple, low-level mechanism. Thus, applying them to complex tasks requires the resolution of a great many issues. These include problems of knowledge representation, definition and implementation of inference mechanisms, and control of the inference process. This presentation will identify some of the particular issues to be addressed, as well as some possible approaches and early results in ANN-based knowledge processing.

NEURAL NETWORKS

AND

ARTIFICIAL INTELLIGENCE

Claude A. Cruz
Plexus Systems
10 Whitford Road
Nashua, NH 03062
(603) 595-2334

Overview

Project goal: explore *flow-of-activation networks* (FAN) as a new way to represent and process information:

- Inspired by function and organization of biological nerve nets.
- Use many simple processors to collectively perform operations (rather than one, or a few, processors operating independently).
- System performs "pattern processing" operations, rather than "symbolic" or "numeric" processing.
- FAN's can be *adaptive*; network behavior can change over time.
- Simple processors act as "smart memory"— system's processors and memory are combined.
- A FAN is a static network of "nodes" (processors) and "links" (communication paths). Acts like a "circuit" performing some function, rather than a sequence of procedure calls and data-structure instantiations.

GENERAL COMPUTATIONS

DYNAMIC COMPUTATION NETWORKS

- PROCESSES AND DATA–FLOW PATHS ARE CREATED AND DESTROYED DURING RUN

STATIC COMPUTATION NETWORKS

- PROCESSES AND DATA–FLOW PATHS ESTABLISHED BEFORE RUN, UNALTERED DURING RUN

DATA–FLOW NETWORKS

- PASS ARBITRARY STRUCTURED DATA OVER PATHS BETWEEN ARBITRARY PROCESSES

FAN NETWORKS

- POINT–TO–POINT PATHS XMIT ONLY SCALAR DATA (NO DATA STRUCTURES)
- PROCESSORS OPERATE ON SCALAR LOCAL DATA (e.g. ACTIVATION LEVELS)

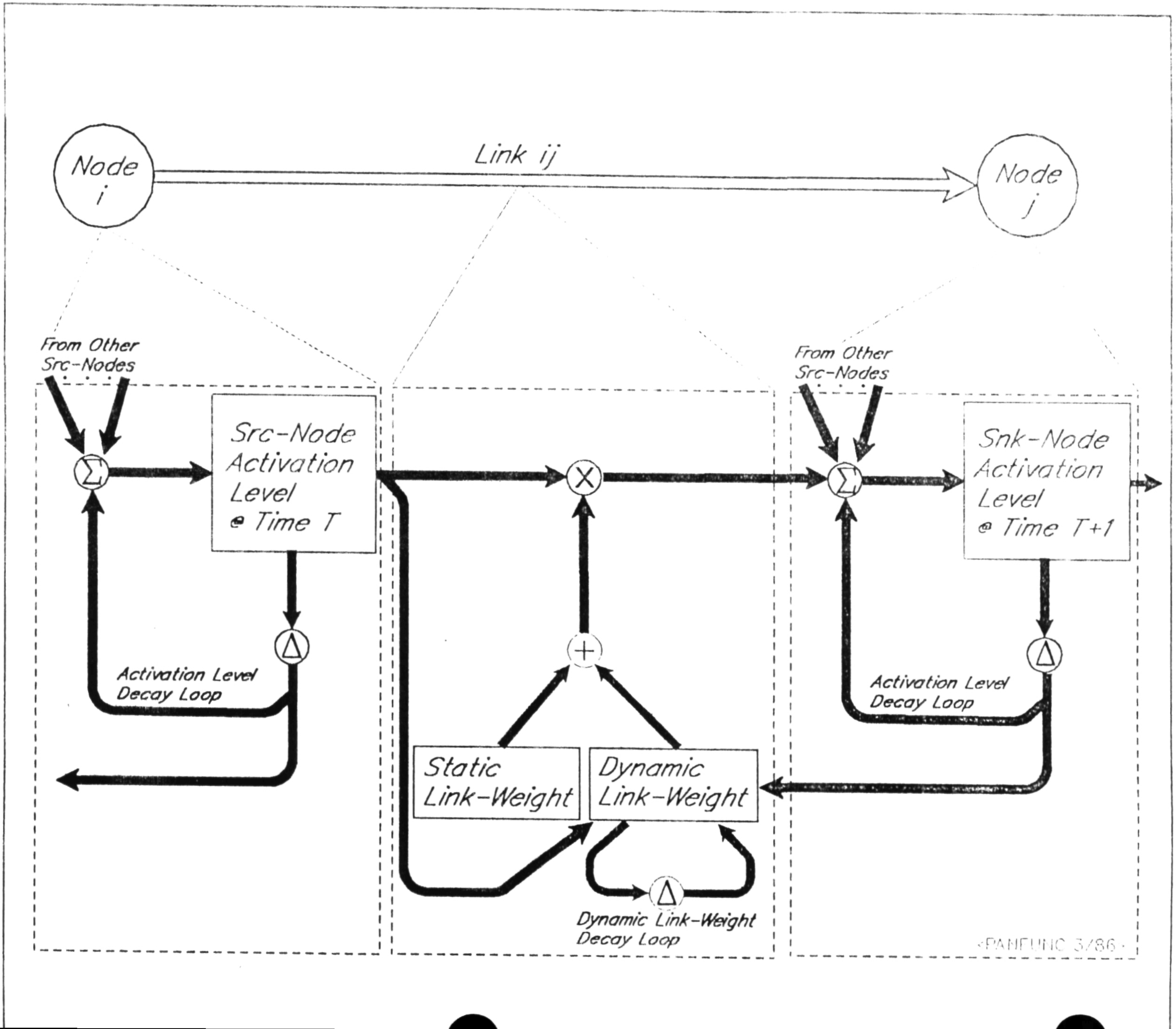
PAN NETWORKS (NEURAL NETS)

- NODE PROCESSORS COMBINE LOCAL STATE INFORMATION
- LINK PROCESSORS XMIT MODIFIED STATE INFORMATION

PANO...

- PARTICULAR NETWORK TOPOLOGY AND NODE/LINK PROCESSOR CHARACTERISTICS

<NETTAX 2/87>



Link ij
Node i
Node j
From Other Src-Nodes
Src-Node Activation Level @ Time T
Activation Level Decay Loop
Static Link-Weight
Dynamic Link-Weight
Dynamic Link-Weight Decay Loop
From Other Src-Nodes
Snk-Node Activation Level @ Time T+1
Activation Level Decay Loop
<PANFUNC 3/86>

Computing by flow-of-activation

- A non-von Neumann computing mechanism
- FAN attributes:
  - Highly parallel (fast)
  - Simple, uniform (pattern) processing
  - Can be adaptive ("learning" model)
- Basic bahavior:
  - Static net represents events and responses
  - Events "activate" nodes
  - Nodes drive other nodes by "flow-of-activation"
  - Active nodes trigger actions
  - Result: each event and each action processed simultaneously in parallel

Some Attributes of Natural Intelligence

- Purpose: to enable an organism to meet its needs (goals) in the face of a changing environment

- Learning is key; "hard-wired", "programmed" behavior limits organism's adaptiveness (like "brittle" programs)

- Nature of environment:

    - Contains regular "features" (entities and events)

    - Entities obey certain laws

    - Both features and laws exhibit some variability

- Intelligence capitalizes on these traits:

    - Learn to recognize features, and to associate significance ("meaning") with them

    - Predict attributes and behavior of features— build "world model" of environment

    - Use "fuzzy processing" to make time-varying best guesses about current contents of environment; seek adequate (not necessarily optimal) plan for meeting goals

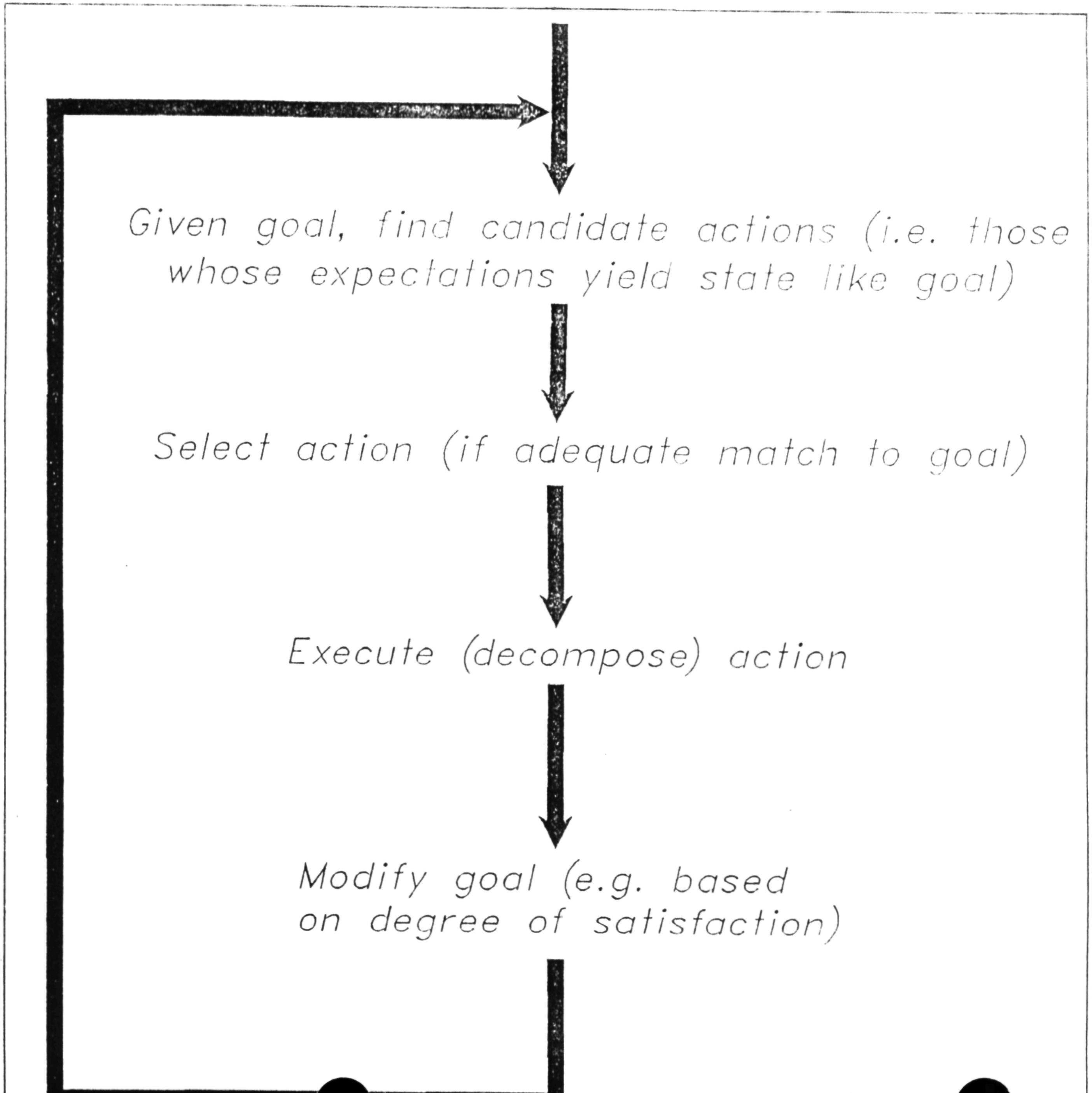
System Embedded in Environment

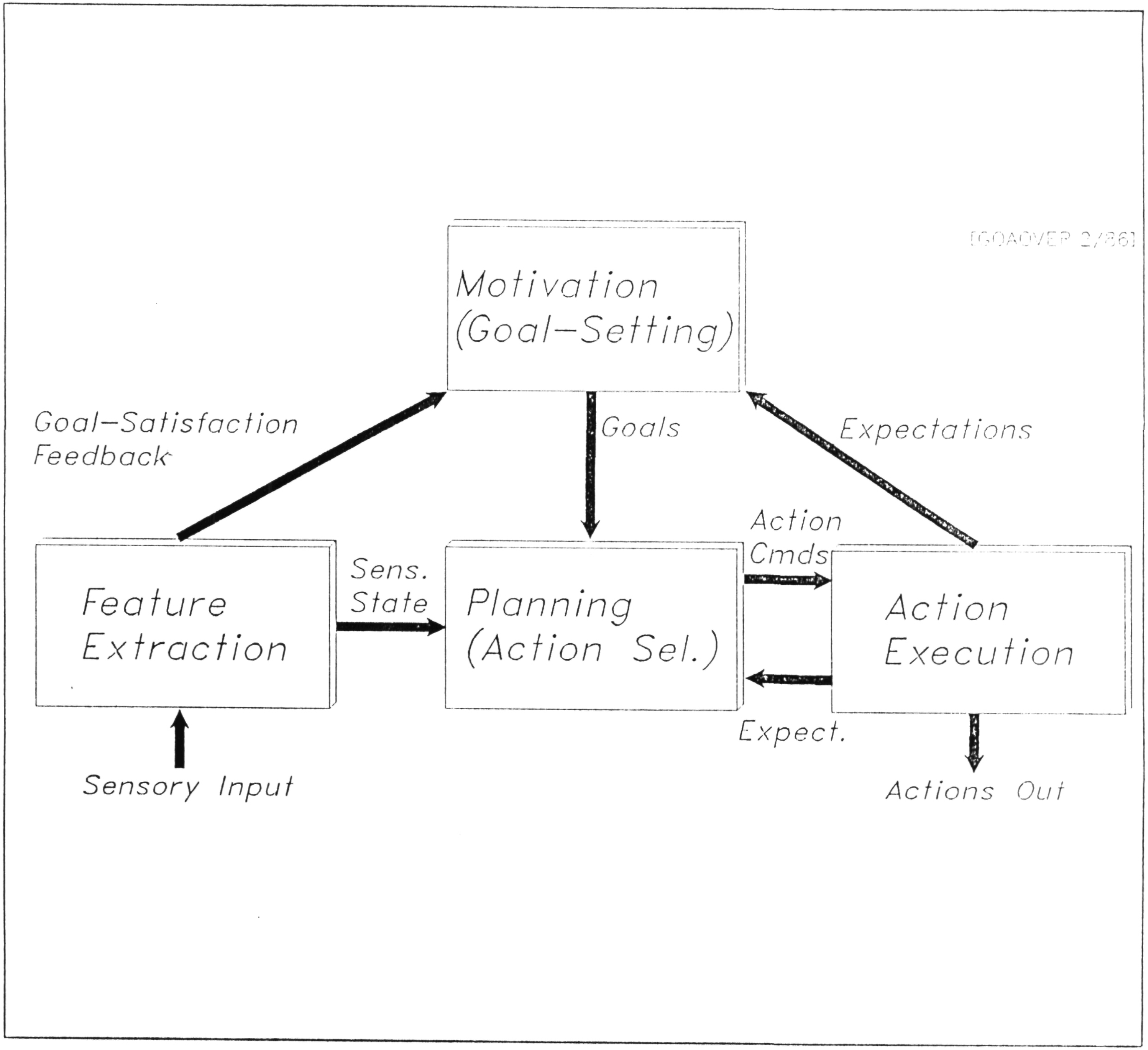
- Cycle relating system and environment:

  1. Event occurs in environment (feature appears);
  2. System detects external event (internal event occurs);
  3. Internal event triggers system response;
  4. Response causes changes in environment (external events).

- System needs internal representation of external reality:

  - "Features" to represent external entities
  - "Relationships" to represent regularities between entities
  - "Operations" to embody responses to external events
  - "Rules" to associate specific responses with specific external or internal events


Given goal, find candidate actions (i.e. those whose expectations yield state like goal)
Select action (if adequate match to goal)
Execute (decompose) action
Modify goal (e.g. based on degree of satisfaction)


[GOAOVEP 2/86]
Motivation (Goal-Setting)
Goal-Satisfaction Feedback
Goals
Expectations
Feature Extraction
Sens. State
Planning (Action Sel.)
Action Cmds
Action Execution
Expect.
Sensory Input
Actions Out

Establishing Representations

- Purpose: set up equivalences (mappings) between external and internal entities:
    - External entities have regularity (content)
    - Presence of entity constitutes an event
    - Internal events produced by "detection" of external events (through fixed "transduction" mapping)

- Two basic schemes are possible:
    - *"Symbolic"* representation: assign a "symbol" (referent) to "stand for" (point to a description of) represented entity. This scheme separates the definition (meaning) of an entity from its referent, coupling the two through an (arbitrarily assigned) association. No relationship (e.g. transduction) required between content of external entity and form of internal representation pointed to by symbol.
    - *"Semantic"* representation: map the "transduction" of an external entity onto an internal "analog". This maps content of external event onto structure of internal representation. The mapping is fixed by transduction process (not arbitrary association). Mapping replaces use of referent (pointer).

Standard AI Techniques

Based on "*symbolic processing*":

- Processing uses *descriptions* (formal models) , rather than *analogs*— incomplete information

- Processing based on *algorithms* (sets of explicit, detailed rules)— missing rules problematic, no generalization possible

- Data content (meaning) not used in processing; formal manipulation only— only programmer has knowledge, or "interpretation", of data's "meaning"

- Through above, processing uses only explicit relationships— cannot use implicit relationships, such as "similarity" between data

- Through above, "learning" (i.e. changes in knowledge-base) requires mediation by centralized "performance assessor"— inherently a serial, high-level function

- Processing is normally *precise*, as are symbols— no role for ambiguity, like that in fuzzy sets or approximate reasoning methods

- Control/decision-making normally centralized—parallel processing difficult, therefore slow

Intelligence via Artificial Neural Systems

Based on *"semantic processing"*:

- Processing based on *analogs* of entities being reasoned about- – acquired through learning or detailed formal specification, thus can contain complete information

- Processing based on structure (contents) of knowledge-entities (KE's), plus connections (relationships) between them— inter-KE connections cause "rule-like" behavior. Generalization possible through "approximate" satisfaction of rules

- Data content (therefore structure) constrains inferences a KE can participate in— uniquely determines "interpretation" of KE's "meaning". No formal manipulation, just inference via "flow-of-activation"

- No implicit relationships, just explicit ones (through inter-KE connections)— "similar" KEs have similar (largely-shared) structure

- "Learning" occurs as local, low-level function (through change in inter-KE coupling)— no central

"performance assessor" needed (though one may be used; e.g. an attention mechanism)

- Both processing and KE's are normally imprecise (but can be made arbitrarily precise)— approximate reasoning is the norm

- Decisions made through locally-controlled flow-of-activation— processing is inherently parallel and fast (but can be made serial)

What Are Knowledge Representation Networks (KRN)?

- An attempt to emulate some biological information processing capabilities ("intelligence"):
  - Process multi-modal input (sensory processing)
  - Produce flexible output activity (distributed motor control)
  - Intelligent decision-making (cognitive functions; adaptive planning using a learned "world model")
- Assumption: choice of low-level implementation medium (FAN) is important:
  - Must encompass above three types of functions
  - Underlies important functional capabilities (associative storage, learning, distributed processing and memory, etc.)
  - Draw insight from biological organizational principles

KRN ARCHITECTURE

- Basic **knowledge units** (**features**, **operations** and **relationships**) are represented by **KRN networks**

- **Inferences** are performed through **interactions** between **knowledge units**

- Underlying **KRN has static structure** (no additions or deletions of nodes or links during inferences)

- Channelled flow of activation creates **dynamic knowledge structures** as subsets of overall KRN (like instantiation of variables)

- **PAN is ideal implementation medium** for KRN (parallel network, flow of activation, learning model)

KRN Architecture

- Overall structure of KRN net— a "feature" hierarchy cross-coupled with an "operation" hierarchy (like advanced nervous systems):

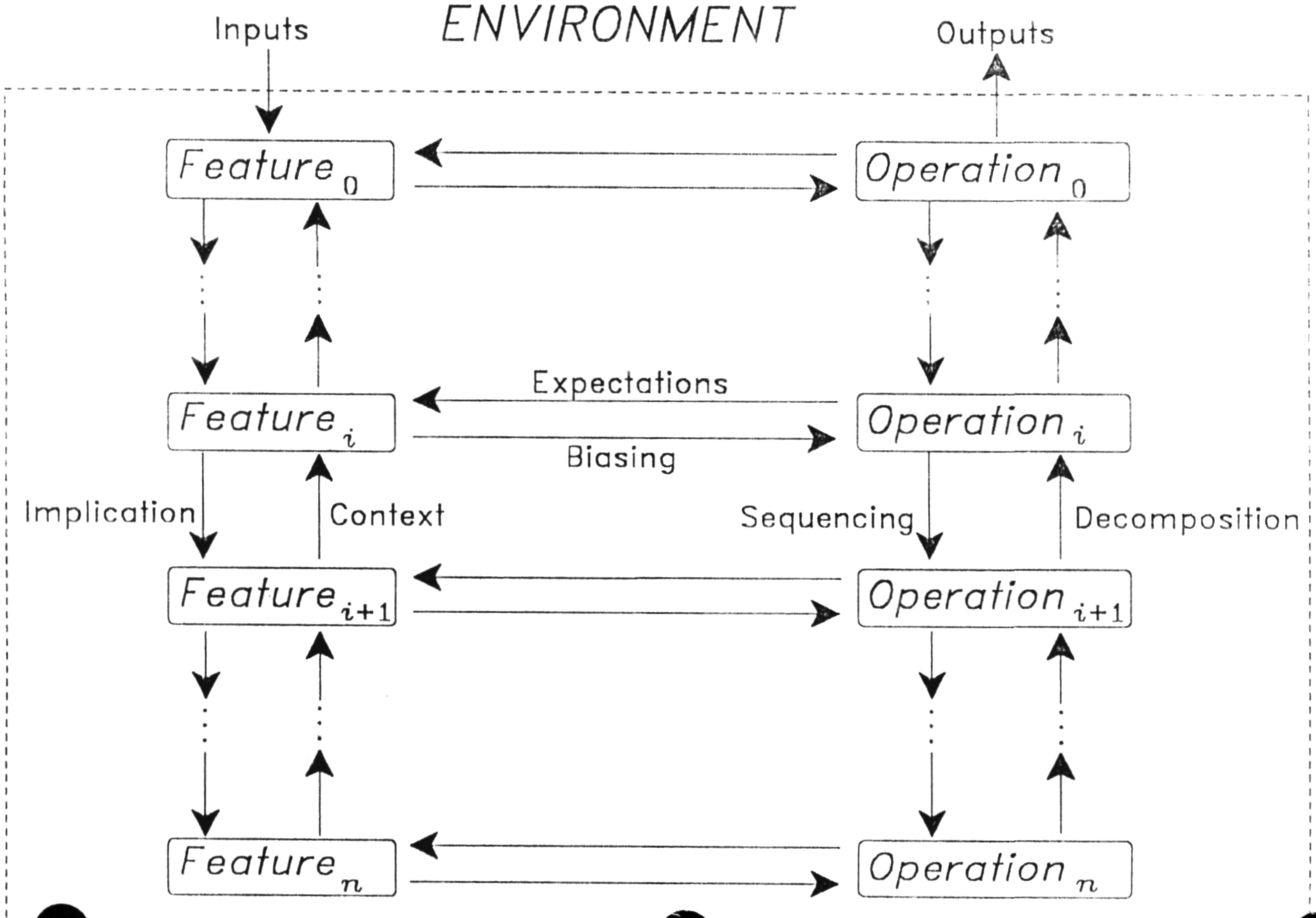
  - Feature hierarchy represents events and entities which the KRN net can recognize. Conjunctions of features define higher-level (more complex) features.

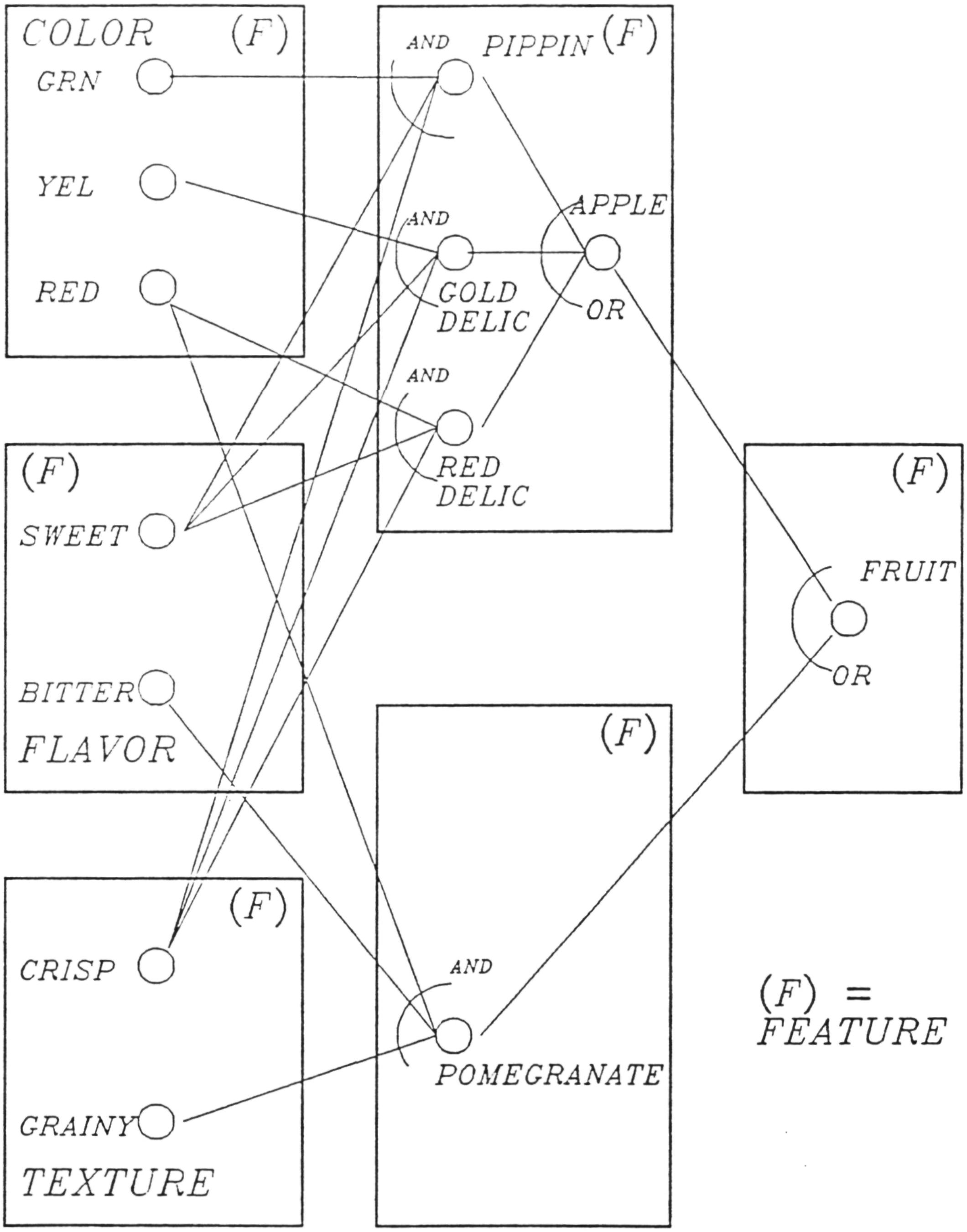
  - Operation hierarchy represents operations (actions) which the net is capable of executing in response to detected features. Operations are decomposed into sequences of sub-operations.

  - Cross-coupling from features to operations provides "rule-like" firing of operations as triggering features become active.

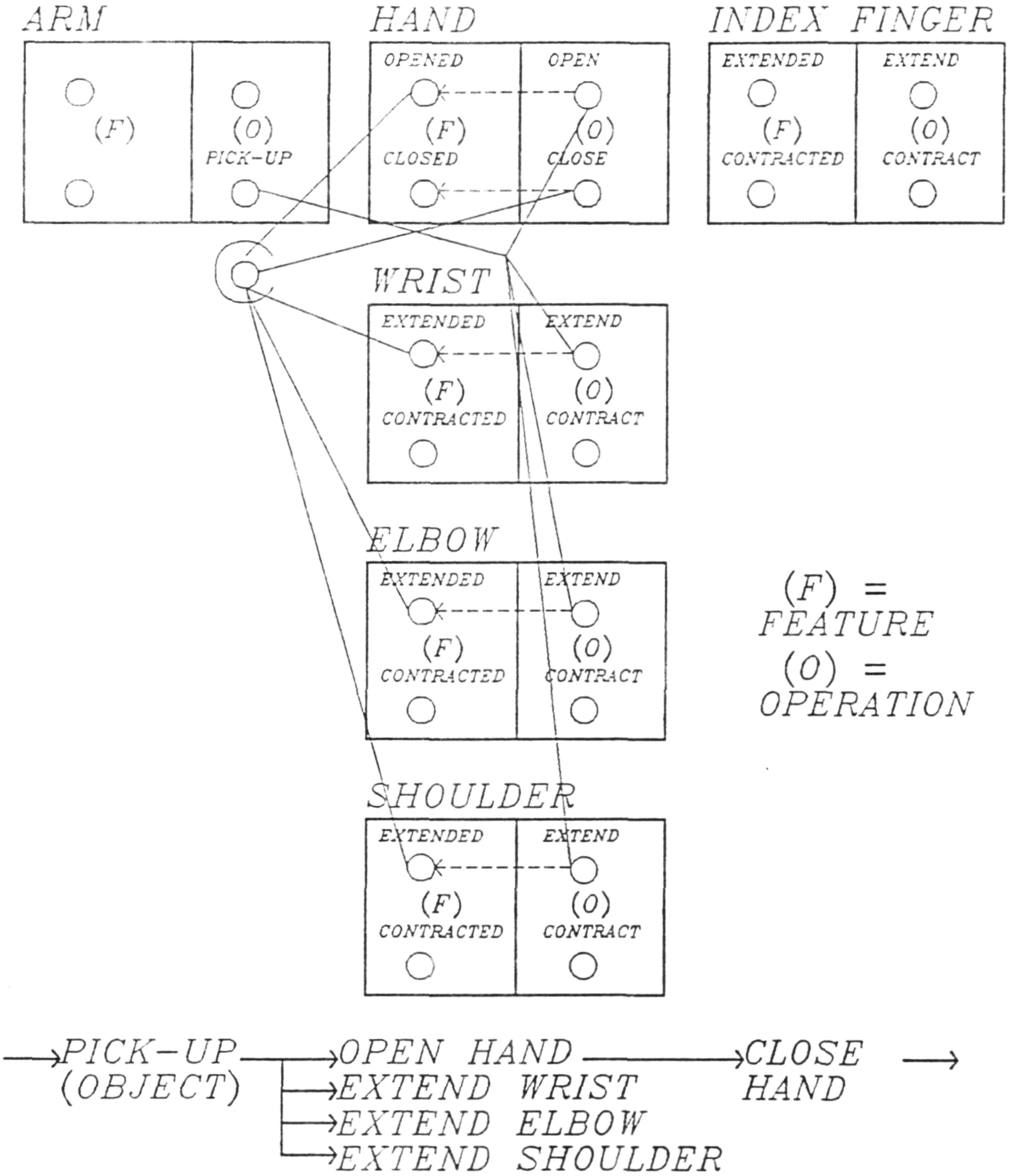
  - Cross-coupling from operations to features activates "expectations" of state-changes which should result from execution of operations. This allows operations to execute in "closed-loop" mode.

KREV0287

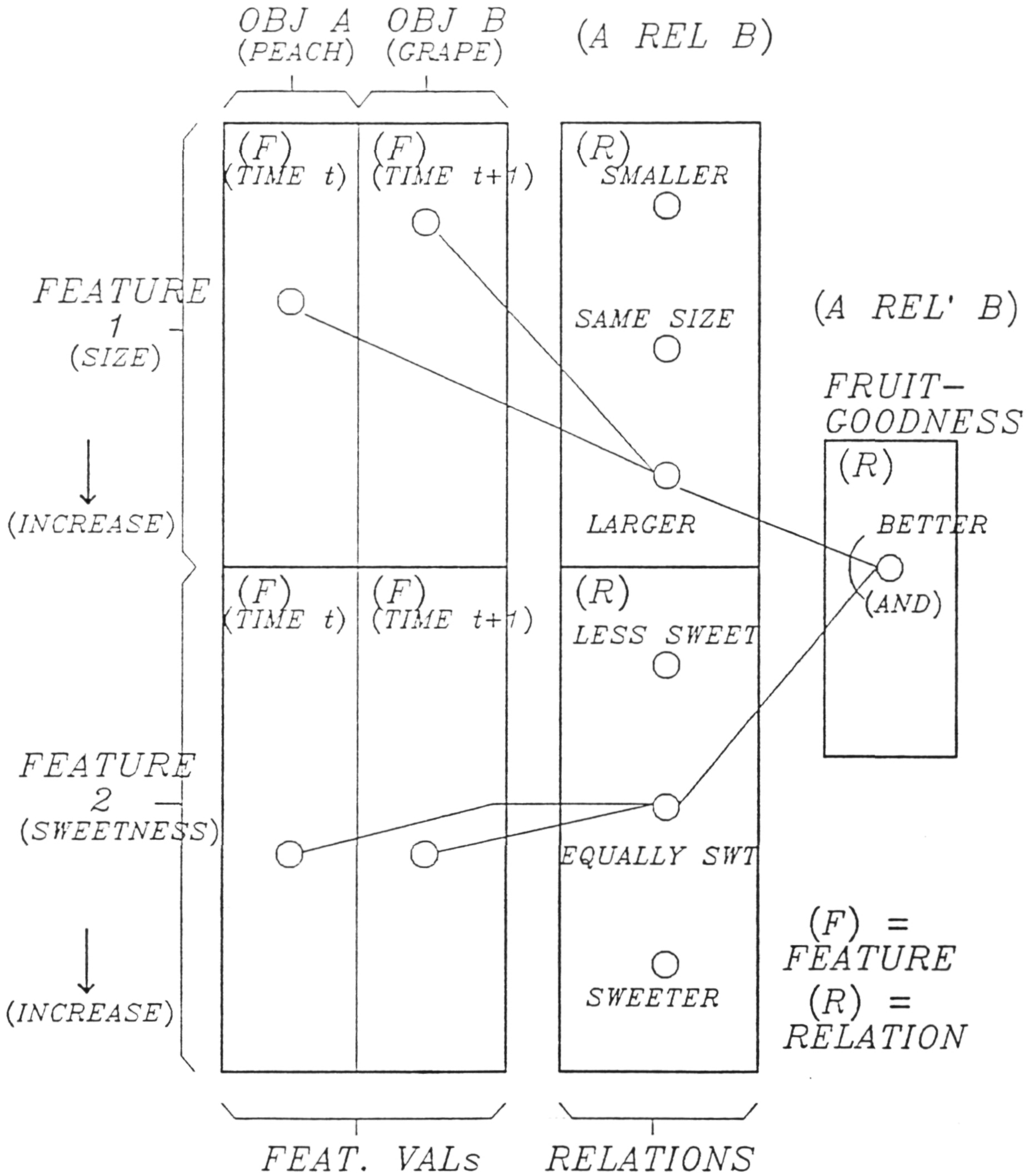

ENVIRONMENT
Inputs
Outputs
Feature $_0$
Operation $_0$
Expectations
Biasing
Feature $_i$
Operation $_i$
Implication
Context
Sequencing
Decomposition
Feature $_{i+1}$
Operation $_{i+1}$
Feature $_n$
Operation $_n$



COLOR
(F)
GRN
YEL
RED
AND
PIPPIN
(F)
AND
APPLE
GOLD DELIC
OR
AND
RED DELIC
(F)
SWEET
BITTER
FLAVOR
(F)
FRUIT
OR
(F)
(F)
CRISP
AND
POMEGRANATE
GRAINY
TEXTURE
(F) = FEATURE



ARM
(F)
(O)
PICK-UP
HAND
OPENED
OPEN
(F)
(O)
CLOSED
CLOSE
INDEX FINGER
EXTENDED
EXTEND
(F)
(O)
CONTRACTED
CONTRACT
WRIST
EXTENDED
EXTEND
(F)
(O)
CONTRACTED
CONTRACT
ELBOW
EXTENDED
EXTEND
(F)
(O)
CONTRACTED
CONTRACT
SHOULDER
EXTENDED
EXTEND
(F)
(O)
CONTRACTED
CONTRACT
(F) = FEATURE
(O) = OPERATION
PICK-UP (OBJECT)
OPEN HAND
EXTEND WRIST
EXTEND ELBOW
EXTEND SHOULDER
CLOSE HAND


OBJ A
(PEACH)
OBJ B
(GRAPE)
(A REL B)
(F)
(TIME t)
(F)
(TIME t+1)
(R)
SMALLER
SAME SIZE
LARGER
FEATURE
1
(SIZE)
(INCREASE)
(A REL' B)
FRUIT-
GOODNESS
(R)
BETTER
(AND)
(F)
(TIME t)
(F)
(TIME t+1)
(R)
LESS SWEET
EQUALLY SWT
SWEETER
FEATURE
2
(SWEETNESS)
(INCREASE)
(F) =
FEATURE
(R) =
RELATION
FEAT. VALs
RELATIONS

Why Build KRN's from PAN's ?

- Need to represent *events* (features and operations):
  - Events are discrete and uniquely-identifiable
  - Represent events by distinct PAN nodes
  - Detection of feature has a "certainty" measure (strength of belief); execution of operation has a "priority" measure (degree of appropriateness)
  - Encode in (a pattern of) continuous-valued PAN node activation levels
- Need to represent *relationships between events*:
  - Events favor or inhibit other events with some pair-wise "degree of causality" (implication strength)
  - Represent events by distinct (positive or negative) PAN links

- Adaptiveness *alters beliefs about degree of influence* of one event on another:

    - ♦ Modify pair-wise degree of causality

    - ♦ Use FAN-link "learning" mechanism (e.g. Hebb model)

- Operations produce *sequences of events*:

    - ♦ Arbitration between candidate operations requires comparison of candidates' "appropriateness" measure

    - ♦ Encode in FAN node activation levels

    - ♦ Temporal constraints important in executing some operations (use network dynamics)

    - ♦ Event sequencing requires "handshaking" (use directed FAN flow of activation)

- Basic functions to *implement behavior* (stimulus-response cycle):

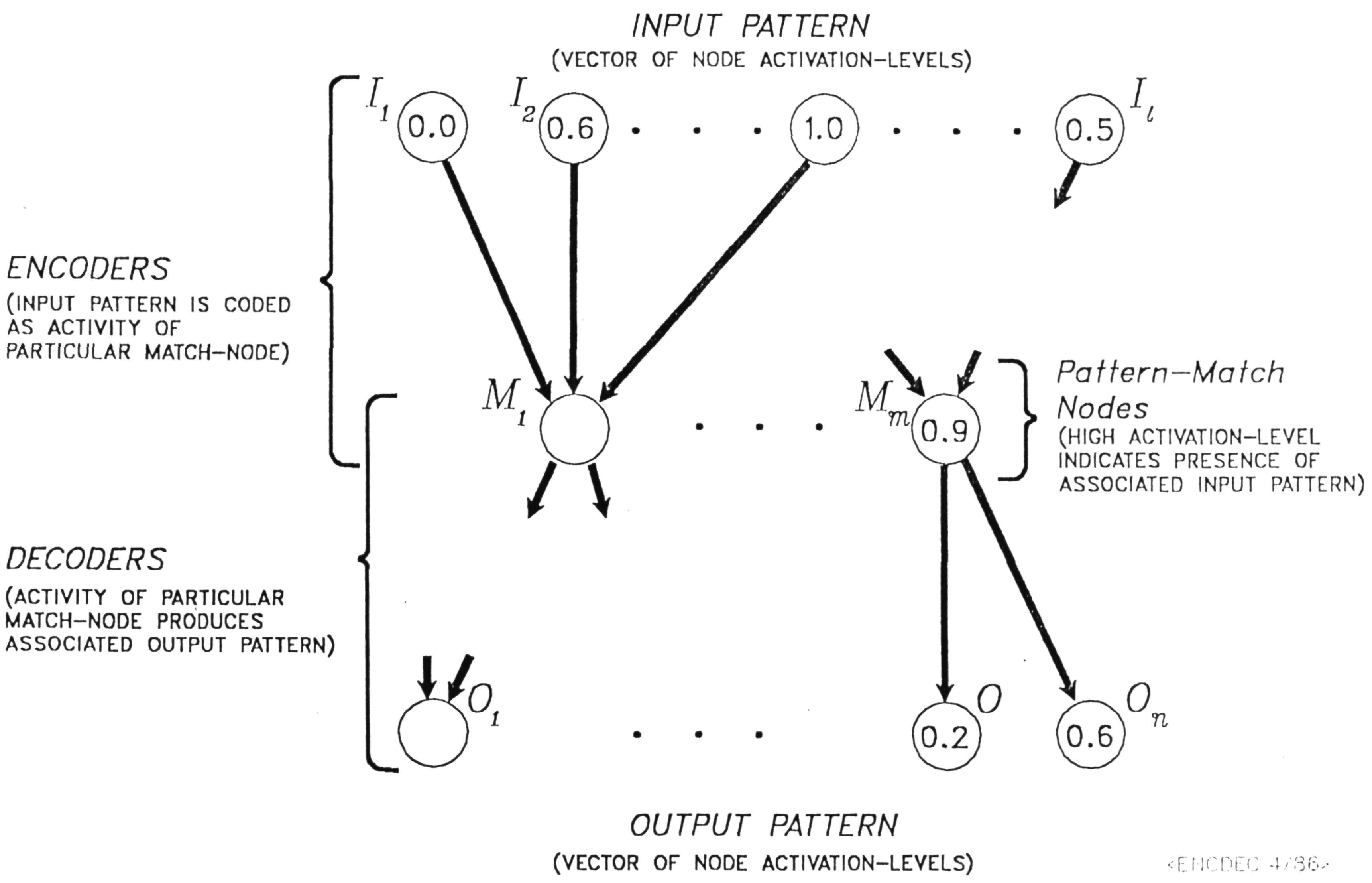
    - ♦ Detect occurrence of events (like IF-clauses; use FAN "pattern-encoder" circuits)

- Internal events represent responses (like THEN-clauses; use FAN "decoder circuits" to produce activity patterns)

- Events and operations may *occur independently*, and should be processed independently:

  - FAN nodes and links are asynchronous independent processors

  - FAN processing is local (events and operations in FAN nodes, inter-node dependencies through FAN links)



INPUT PATTERN
(VECTOR OF NODE ACTIVATION-LEVELS)
I1
0.0
I2
0.6
1.0
0.5
Il
ENCODERS
(INPUT PATTERN IS CODED AS ACTIVITY OF PARTICULAR MATCH-NODE)
M1
Mm
0.9
Pattern-Match Nodes
(HIGH ACTIVATION-LEVEL INDICATES PRESENCE OF ASSOCIATED INPUT PATTERN)
DECODERS
(ACTIVITY OF PARTICULAR MATCH-NODE PRODUCES ASSOCIATED OUTPUT PATTERN)
O1
O
0.2
0.6
On
OUTPUT PATTERN
(VECTOR OF NODE ACTIVATION-LEVELS)
<ENCDEC 4/86>

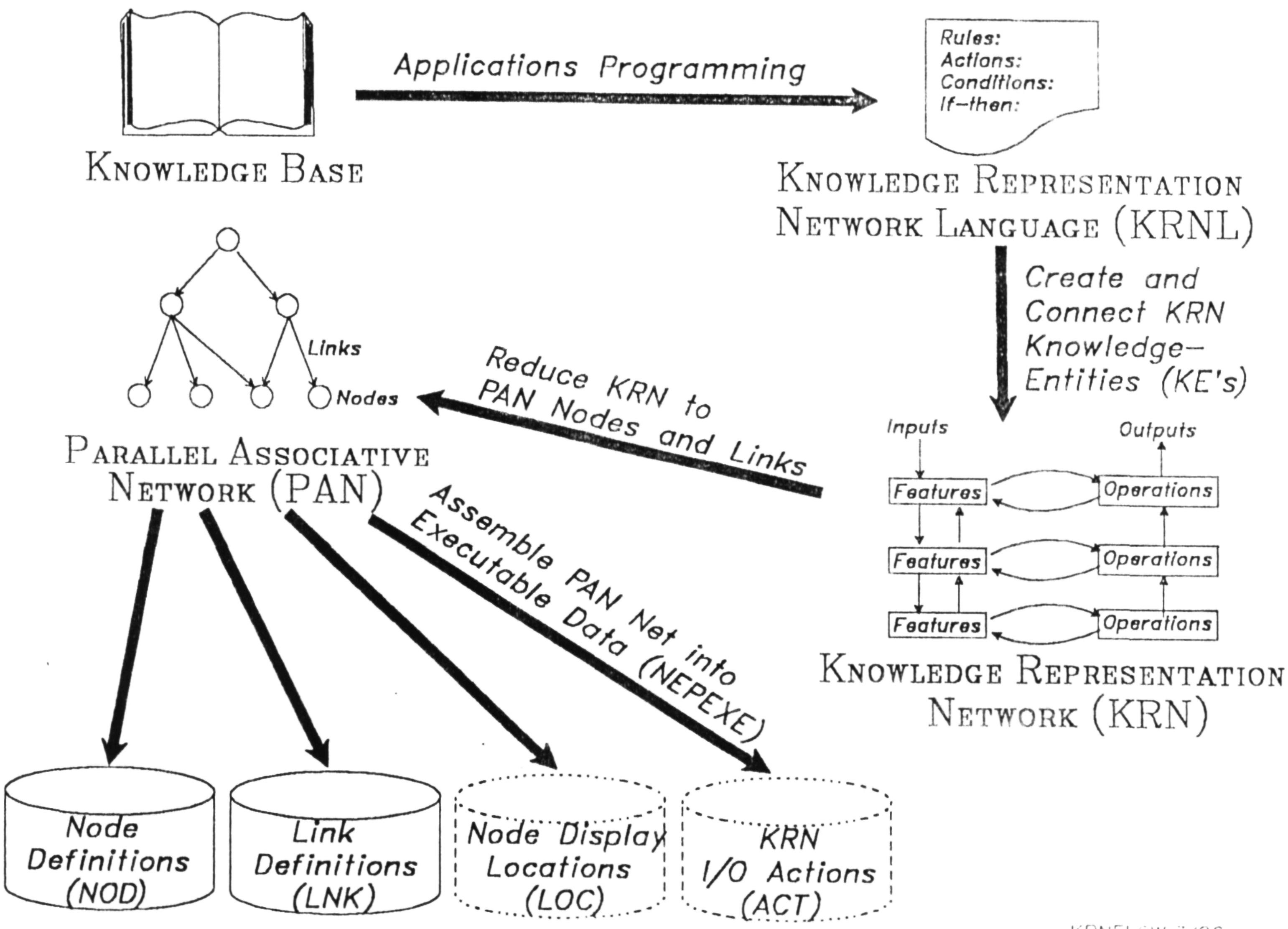
Basic KRN "Building-Blocks"

- Features are represented by FAN "encoder" circuits:
  - Output becomes active IF all required input-feature values are active (present)
  - KRN net inputs can drive encoder input features.
  - Match performed by encoder is "fuzzy"— there is some tolerance for input features within a satisfactory range, not just one exact value.
  - Sets of encoders may take input from same set of input features. Inter-encoder competition is used to maximize activity of single encoder with best match to current inputs.
  - FAN "column" circuit acts as encoder; FAN "hypercolumn" circuit acts as competing set of encoders with shared inputs.
- "Decoders" produce a given output activity pattern across a set of output nodes:
  - In producing specific network state-transitions, decoders act like production-rule THEN-clauses.

- Decoder input activity level scales output activity vector.
- Decoder outputs can produce output from the overall KRN net.

- Encoders cascaded into decoders form "fuzzy" state-machine:
  - Encoders detect current network state;
  - Active decoders trigger associated decoders (via FAN links);
  - Active decoders cause state-transition (i.e. change activity of features which feed encoders).

- Local FAN "attention mechanism" can be used to "enable" or "disable" sets of encoders and decoders. This helps focus processing. (Use and implementation of attention is crucial current research area).



Applications Programming
KNOWLEDGE BASE
Rules:
Actions:
Conditions:
If-then:
KNOWLEDGE REPRESENTATION NETWORK LANGUAGE (KRNL)
Create and Connect KRN Knowledge-Entities (KE's)
Links
Nodes
Reduce KRN to PAN Nodes and Links
PARALLEL ASSOCIATIVE NETWORK (PAN)
Inputs
Outputs
Features
Operations
Features
Operations
Features
Operations
Assemble PAN Net into Executable Data (NEPEXE)
KNOWLEDGE REPRESENTATION NETWORK (KRN)
Node Definitions (NOD)
Link Definitions (LNK)
Node Display Locations (LOC)
KRN I/O Actions (ACT)

<KRNFLOW 3/86>

NOTES

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

S18-61

153/8

N91-71368

Willard R. Taber, Ph.D.

General Dynamics
San Diego, California

Dr. Taber majored in physics and astronomy at the University of Iowa, receiving his B.S. in 1973. Following graduation, he worked in the field of soil physics and chemistry, writing computer programs to identify ecologically important variables for nuclear reactor sites. He also began work in computer science at Texas A&M on medical imaging at the College of Medicine. From 1978 through 1984, Dr. Taber designed and built a small computer, performed hardware maintenance on Interdata minicomputers at the chip level, and integrated computers with laboratory equipment, such as spectrophotometers. A major part of his work involved building image and signal processing software. He graduated from Texas A&M in 1984, joining General Dynamics shortly thereafter.

FUZZY LOGIC OPERATORS AND NEURON ACTIVATION FIELDS

Abstract

A neural structure in light of fuzzy sets and operators is examined. During a study of underwater acoustic signatures, it was discovered that a simple version of the avalanche could be improved for classification purposes by adding two simulated hardware memories (or latches) to each neuron. The performance of the new structure, called a neuron ring, approximates the performance of cross correlation. Only simple operators, such as the sigma-count and the triangular norms MAX and MIN are necessary. In brief, the pattern exciting the neuron is viewed as simply a means to induce a possibility distribution in the neuron. The height of the distribution is partial support for the hypothesis in question. The summation of support after a time sequence of excitation is support for the hypothesis.

Fuzzy Sets and Neural Networks

W.R. Taber and R.O. Deich

General Dynamics, Electronics Division, Box 85310
San Diego, CA 92138, MZ 7202-K

INTRODUCTION

Fuzzy sets [15,17] and their operators have interesting applications in neural networks. The performance of fuzzy decision rules for pattern classification provides a compelling reason to use graded set membership functions. As we will show, fuzzy operators enable a simple structure, the neuron ring, to classify non-stationary patterns in the presence of severe noise. Without fuzzy processing, the ring is exceptionally sensitive to noise. Its operation as a shift invariant filter is not appropriate.

The underlying problem with non-fuzzy rings is that decision rules examine terminal activation instead of the historical activation record.

In this paper, we report the performance of neural structures trained with undersea ship signatures from San Diego Bay and elsewhere. We compare performance to cross correlation - a conventional processing algorithm that seldom fails. Backpropagation [9,10,13] performance is also compared both with and without stationarity assumptions.

Reference to ship signatures should not sidetrack the reader from recognizing the contributions of fuzzy processing to pattern recognition. Fuzzy processing may be the best model for non-stationary patterns - those patterns that change their descriptive statistics over time[1]. That ship acoustic energy survives to propagate over geographic distances is amazing. Yet the digital computer, simulating fuzzy rings, correctly identifies ships in the presence of pseudo-white, colored, chirp, and other types of noise. This being the case, the algorithm outlined in this paper has intrinsic value apart from its underwater application.

FUZZY SET EXAMPLE

The indicator function of a standard set is either a 1 or a 0. Either the set element is present or it is not. In distinction, the indicator function of a fuzzy set admits graded membership. An element can be present or absent, or it may be present to a degree. A simple example will illustrate this concept.

Suppose a man with a full head of hair is the subject of an experiment to quantify baldness. The experiment consists of plucking a single hair then recording the answer to the question, "Is this man bald?" The first time, the answer is definitely "no". However, continued experiments with the same subject and other observers will lead to positive answers. The outcome of the experiment is justification for statements of the form "the probability that this man is bald is .50."

Are there other ways to estimate baldness? The answer is yes, and fuzzy set theory provides an approach.

A fuzzy practitioner views bald individuals as a set and tries to estimate an individual's membership from data. For example, he estimates the number of hairs on the average head, then estimates the number of hairs on the subject's head. Without solicitation, he is able to use an estimate of the

---

[1] See reference 2 for definitions.

number of hairs as the numerator of the ratio of the subject's hair to the average. This is an estimate of the approximate baldness degree. His membership in the set of bald individuals is about the ratio.

Often, there is no information or even requirement to justify probability estimation either from a frequentist's or from a Bayesian's viewpoint. It is in these situations that fuzzy sets offer complementary value.

THE NEURON RING

There are a number of connection intensive networks for pattern classification [7]. The Grossberg avalanche [5] cascades neural elements to learn and recognize spatiotemporal patterns. In signal processing terminology, the avalanche recognizes nonstationary patterns. Hecht-Nielsen [6] further reduced the connectivity of this structure in the commercial SPR (spatiotemporal pattern recognizer) feedforward network.

The neuron ring resembles the SPR but has new architectural features. Its closest approximation is the torus of Goles [4]. The ring's processing element is the DPNL- the dot product neuron with latches that hold time (T) and activation (M) values. These are visible in figure 1.

The DPNL operates in the following manner. During training, the reference pattern is hardwired (fast-learn mode) into the neuron as vector Z. During recognition, the test pattern U is dotted with Z yielding a scalar. This quantity initializes an accumulator in DPNL *i*. The next operation multiplies the unit's old activation, X_{old}, by -a. Another term is computed with *decouple* gating the activation of the previous neuron. The activation sum is multiplied by a gain factor A, yielding the activation increment δX_i^{new}.

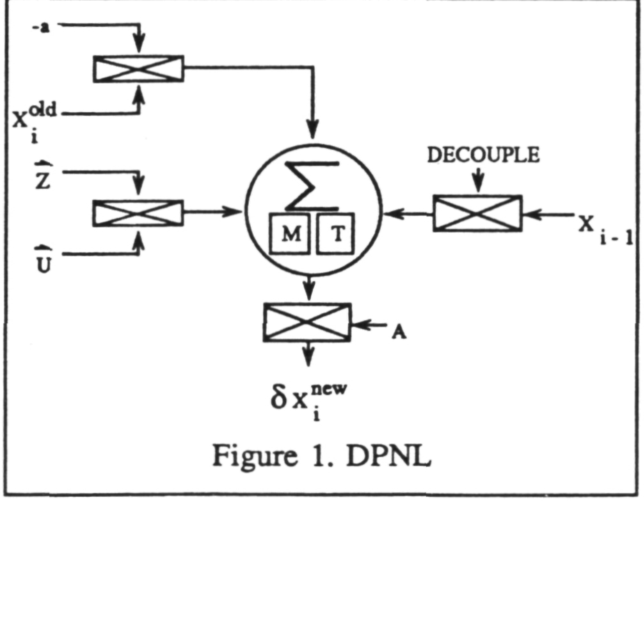


Figure 1. DPNL

Decouple couples a small fraction of activation or *encouragement* forward. When zero, the signal enables a special mode for spectral classification using permutations of firing order as similarity metrics. When decoupled, the DPNLs activate independently, leaving an audit trail of firing order.

Equation (1) relates these quantities for DPNL *i* with a variation in the terminology established by Hecht-Nielsen for the SPR.

Figure 2 illustrates the tertiary structure of a ring assembly. The lateral feedforward and the single return are its gross features. The input layer is not shown.

A pattern sequence excites the ring to a graded activation level. Rather than invoke the all or none firing principle, the DPNL exports its activation untouched except for hard limiting the value to the unit interval [0,1]. The basis for excitation is partial correlation by dot product. That is, correlation at zero time lag. For pattern vectors of unit length, the dot product is in the unit interval. The higher the number, the greater the similarity between V_1 and V_2 on the unit hypersphere in $\Re^n$. Each pattern excites every DPNL. Initially, both latches are reset to 0. As the patterns arrive at the DPNL, the registers latch new values. The max latch, M, changes only when the current activation exceeds M.

After complete pattern presentation, the activation field or max latch array is ana-

lyzed with fuzzy primitives. It is this historical record and processing that is absent in most network paradigms.

$$X_i^{new} = X_i^{old} + A[-a X_i^{old} + b I_1 + c I_2]^+$$

Equation 1

X_i^{new} = Activation for neuron i
X_i^{old} = Old activation
A = Attack factor
a = Decay constant for old activation
b = Gain for activation from previous neuron (decouple)
I_1 = Activation from previous neuron
c = Gain for dot product
I_2 = Dot product of pattern $\vec{Z}_i$ with unknown $\vec{U}_i$

Before discussing fuzzy activation field processing, we first examine two point-sensitive decision rules in common use before this study. They are:

D_1: The ring with the highest acceptable activation in its last neuron wins.

D_2: The ring with the first activation of 1 wins the competition.

We eliminate D_1 immediately. Imagine a ring with 100 neurons. Suppose the test input is identical to the training pattern except that time slice vector 99 is an attenuated version of the true vector. DPNL 100 will not be fully excited. Another ring, conditioned on a different signal, may win at the end of time slice 100 by a simple twist of fate at time 99.

The other rule has competitive merit but it makes little sense when applied in a noisy environment with rampant phase errors. Transients may induce random neuron firing.

Point failure is serious because a system which allows it to occur discounts historical evidence in favor of the current state as does a markov process. Other decision rules are possible. Before addressing the rule(s) of choice, we will discuss some aspects of sampling and phase error.

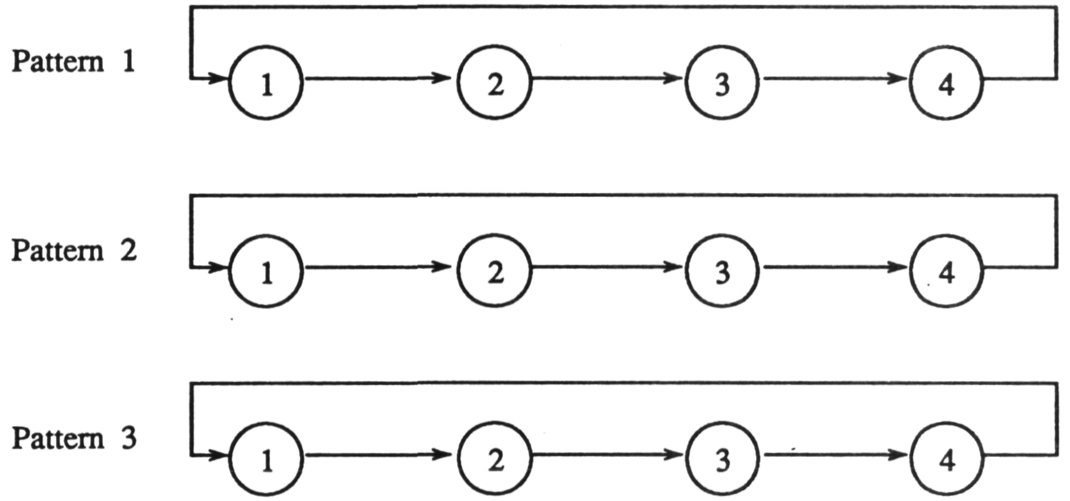


Figure 2: A three pattern ring assembly. Pattern i conditions ring i. Each DPNL supports a microhypothesis (MH) for time j.

Phase error can be illustrated with an example from the undersea application. Assume the acoustic signature is a periodic and deterministic function of propeller angle as it spins. Suppose the training pattern was sampled when the propeller was vertical. The remaining samples follow at equal intervals. During a sea trial, the probability that sampling started with a vertical propeller is small. This being the case, the test and the training signal are similar except for a phase difference.

The re-entrant ring compensates for phase, although the effect is large only for small rings. It makes no difference which DPNL is first stimulated; DPNLs activate around the ring by virtue of the syndetic lines. Phase displacement lies latent in the T latch chain.

A better decision rule or heuristic for pattern classification will now be sought.

Each DPNL continually provides a statistic for testing the hypothesis <u>the pattern is as expected</u> as a correlation by-product. The first DPNL in a ring holds the pattern vector for time slice 1. Therefore, it estimates the grade of membership or suitability of the test pattern's first vector. The second DPNL estimates the suitability of the second pattern vector. Anthropomorphically, operation is a question and answer sequence: "How well, on a unit scale, do you like what you see at this time?" The problem is to decide, that of all the activations generated by a DPNL during presentation, which best indicates support for the hypothesis?

The answer is stated without proof; it is the height of the time series of activation for the DPNL, otherwise called the fuzzy possibility measure. It is just the contents of M.

We estimate the compatibility of the test pattern with the ring as a whole. The appropriate operator is the sigma-count [16] of the fuzzy set M. Kosko [8] proved the sigma-count, ΣC, is a positive measure of set cardinality. It represents support for the ring's hypothesis.

Up to the present time, the discussion has been limited to a single ring. More signals require more rings. The supervisory system, if it exists, recruits empty rings and conditions them as necessary based on mean squared error criteria.

Assuming ΣC_i is support for signal *i*, what rule robustly adjudicates the race for classification?

This discussion argues that no point estimate from the ring <u>during excitation</u> will suffice as a fuzzy statistic, unless its value is unity. We propose the following calculations as a foundation for a better decision rule.

$$\text{Support}_i = \Sigma M_{ij}$$

$$\text{Non-support}_i = \text{Card} - \text{support}_i$$

where support is belief in the hypothesis <u>the pattern is *i*</u>, and *j* is the DPNL index. Card is the cardinality of the ring's non-fuzzy superset, i.e., the number of DPNLs per ring. Non-support is the degree to which the hypothesis is not warranted.

A walk through figure 3 data will clarify the procedure for classification. Morphologically, the assembly that generated the data had 10 rings. Each had 20 DPNLs, one per time slot.

The numbers in the top matrix are the contents of the max latches at the end of the excitation. Rows index the pattern while columns index time.

The contents of the max latch for DPNL for pattern 1, time 18, is 9 (the lack of a decimal is a concession to display technology). The 114 under ΣC is the support for pattern 1 while its non-support is 200 - 114 = 86. With the implied decimal, these values are .9, 11.4, 20, and 8.6.

The certainty ratio (CR) is calculated:

$$CR = Support_i / \Sigma\,(Support_i)$$

Next, each CR is divided by the maximum in the CR column. Finally, "FUZZY MEMBERSHIP" (FM) is reported as a percentage.

The quantities under "FUZZY MEMBERSHIP" indicate the degree to which the training pattern is supported by the data if a decision must be made. Its associated degree of non-support is 100 - itself. For pattern 1, the values are 100 and 0.

If a delayed decision is permitted, the ΣC column is more appropriate than FM. If the support for any ship is lower than a threshold, classification can be deferred until more samples are available.

The formal definition of the possibility computations is as follows. Let the time slot set for a single neuron be A = {1, 2, 3, ..., n}. Let x_i be the activation of a neuron at time *i*:

$$x = \{x_1 \quad x_2 \quad x_3 \ldots \quad x_n\}$$

STATISTICS FOR TEST FILE: SHIPS INPUT SIGNAL: Boat 2

WINDOW =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	ΣC
Boat 2	0	0	2	4	6	5	5	5	4	6	6	7	7	7	6	8	7	$\boxed{9}$	10	10	$\boxed{114}$
Boat 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Elizabeth	0	0	2	4	3	3	3	3	3	2	2	2	2	2	2	2	2	4	5	5	51
SEINER	0	0	2	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	14
FF1041A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FF1041B	0	0	2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	12
FFG41B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FFG41C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DREDGE	0	0	0	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	1	20
ZODIAC	0	0	2	4	5	5	5	4	4	4	4	4	3	3	3	3	3	3	4	6	69

CERTAINTY MEASURES FOR INPUT SIGNAL: Boat 2

TRAINING SIGNAL	CERTAINTY RATIO	FUZZY MEMBERSHIP
Boat 2	$\boxed{0.407}$	$\boxed{100}$
Boat 3	0.000	0
Elizabeth	0.182	45
SEINER	0.050	12
FF1041A	0.000	0
FF1041B	0.043	11
FFG41B	0.000	0
FFG41C	0.000	0
DREDGE	0.071	18
ZODIAC	0.246	61

CLOSEST MATCH FOR INPUT SIGNAL: Boat 2
CURRENT PERTURBATION PERCENT = 40
TEST NUMBER = 25

FIGURE 3. Computer screen depicting the excitation signal Boat 2 with 40 % noise in its power spectrum. Numbers in the top matrix are the contents of the max latch at termination. The last column is the measure of support for the hypothesis implied by the row heading.

Then the possibility distribution:

$$\Pi_X = x_1/1 + x_2/2 + x_n/n$$

is induced on the neuron. The term $x_1/1$ means: the possibility that the signal is appropriate given the signal at time 1 is x_1. Then the possibility measure

$$\pi(A) = \max(x_1, x_2, x_3, \dots x_n)$$

is held in the max latch, M.

Either the ΣC or the FM statistic is now considered a better decision statistic than those used by either D_1 or D_2. We adopt a decision rule:

$$D_3 = \mathrm{arc}(\bigcup_i \Sigma C_i)$$

or

$$D_4 = \mathrm{arc}(\bigcup_i FM_i)$$

where the functor *arc* is a pointer back to the pattern name. Thus $D_3(114)$ is boat 2.

One objective of pattern recognition is to generalize [1] , to go from a specific signal to the class to which it belongs. An admissible algorithm is driven by the joint similarity between the training pattern(s) and the set of all patterns produced by the same signal source. It infers that while the test signal is different from any in the training set, it has the gross properties of, say, a frigate. There are at least two variations on generalization. The first is mentioned only for the sake of completeness.

Variation 1 examines the output string of the supervisory system if present. For example, the ring's postprocessor declares the signal to be the frigate Mir on the basis of its emissions. Variation 1 parses the text string for the underlined word and subsequently declares the class to be frigate.

Variation 2 scrutinizes the output of the ring - the ΣC or FUZZY MEMBERSHIP tal data alone. Figure 3 illustrates that two of the fishing boats excite the ring but that the Zodiac raft does also. These boats have much in common in the frequency domain. On a broader note, the ring permits the testing of any fuzzy hypothesis that can be constructed from the universe of discourse.

Analysis of the activation history or field is facilitated with an element of set theory called the power set - the set of all subsets from the universe of discourse. Its cardinality is 2^n.

Two constructs derivable from the power set are the frame of discernment or disjunctive frame (Strat [11]) and the frame of concernment or conjunctive frame. In all, they contain $2^{(n+1)}-(n+1)$ unique hypotheses and from them, any hypothesis with conjunction/disjunction is constructible. For example, support for the hypothesis:

H(*Elizabeth*)

is 51 in the ΣC column. We can also test:

H(the *Elizabeth* or boat 3)

with MAX, the fuzzy set union operator.

The arithmetic is a straightforward application of the sigma-count, and the Frank [3] triangular norms and co-norms MAX and MIN:

H(*Elizabeth*) = 51

H(*Elizabeth* or boat 3) =
MAX(0, 51) = 51

H(*Elizabeth* or boat 3)

AND
H(FF1041A or FF1041B) =
MIN(12,51)= 12

Yager [14] and Zadeh [15,17] discuss other operators for reasoning with uncertain information. Similar exercises apply to FM.

EXPERIMENTS AND NOISE

Signatures from ten vessels were collected from San Diego Bay with an omnidirectional hydrophone. They joined an extensive library of marine mammal vocalizations, munition launches, seismic explosions, and other acoustic events.

We soon developed a comprehensive procedure for simulating ship encounters and testing performance. Classification merit is equated to the probability of correct classification, PCC. Graphs of this function have PCC on the vertical axis and percent noise perturbation on the horizontal. The graph indicates the sensitivity level of the procedure under test to levels of increasing noise. Noise is added as percentage of signal power from 0 to 100 percent. A noise level of 25% implies there is 25% uncertainty in the true value of any frequency bin.

All signatures had ambient bay noise. They also contained aperiodic impulse spikes from nearby power rails. These were attenuated with a median filter (Taber [12]) before Fourier transformation.

Uniformly distributed pseudo-white noise is not the only kind of noise in the ocean. As an aid to more comprehensive situations, we used five noise types.

- uniform white
- ramp up with frequency
- ramp down with frequency
- time shift
- convex combinations of signals

Uniform noise occurs in narrow band samples. Our passband was 0-3.5 KHz. In some cases, the uniform noise assumption is justifiable. However, for example, the sudden appearance of a second boat in the water injects frequency and range dependent noise.

Ramp up and ramp down are analogous to linear chirp in radar. The amount of noise is frequency dependent; the amplitude of the zero mean noise either increases or decreases with frequency, simulating ocean anomalies and opening and closing ranges.

Time shifts are expected. Seldom will the training signal be an exact replica of the test. An ocean buoy for monitoring harbor traffic must contend with tracking vessels over a range of perhaps hundreds of miles. In the laboratory, we simulate the buoy by taking the test signal from a different tape or tape segment than the training signal.

Finally, convex combinations of existing signals test the resolving power of the network to identify fleets of ships. Can it generate activation consistent with known signal mixes? For example, if we mix 75% destroyer with 25% frigate, can the network confirm the 75/25 split? Convexity means that for any combination C, of signals, S_k,

$$C = \alpha S_1 + \beta S_2 + \gamma S_3 \ldots,$$

the coefficients sum to 1.

RESULTS

Our experiments tested the ring's ability to recognize signals in the presence of severe noise. Figures 4 and 5 illustrate the

varied behavior of the algorithms in uniform zero mean noise. Cross correlation almost always recognizes the signal. Backpropagation (BP) trained with the average Fourier power spectrum rather than the entire non-stationary spectrum does not perform adequately. Its failure justifies the use of the non-stationary signals for training in spite of the expensive training sweeps. The BP network for figure 4 has 16 input neurons, 9 middle layer neurons, and 5 in the output layer (16:9:5). The BP networks for figures 5 and 7 are 320:20:5. BP training time on a Sun 3/140 for ten non-stationary signals is approximately 30 minutes. Training time for the ring is less than a second if training time is measured from the time a pattern is interned to the time the network is prepared to recognize signals.

The non-fuzzy rules make the ring into a very narrow bandwidth matched filter for uniform noise. It does not matter whether D_1 or D_2 is selected; the results are similar to the trace in figure 4b.

The ramp up and ramp down tests caused a single fault in the PCC graph. All plots for the fuzzy ring were constant at 100% PCC for noise levels that started at 25% in the 0-60 Hz band and escalated to 75% in the 3.5 KHz band, and constant for

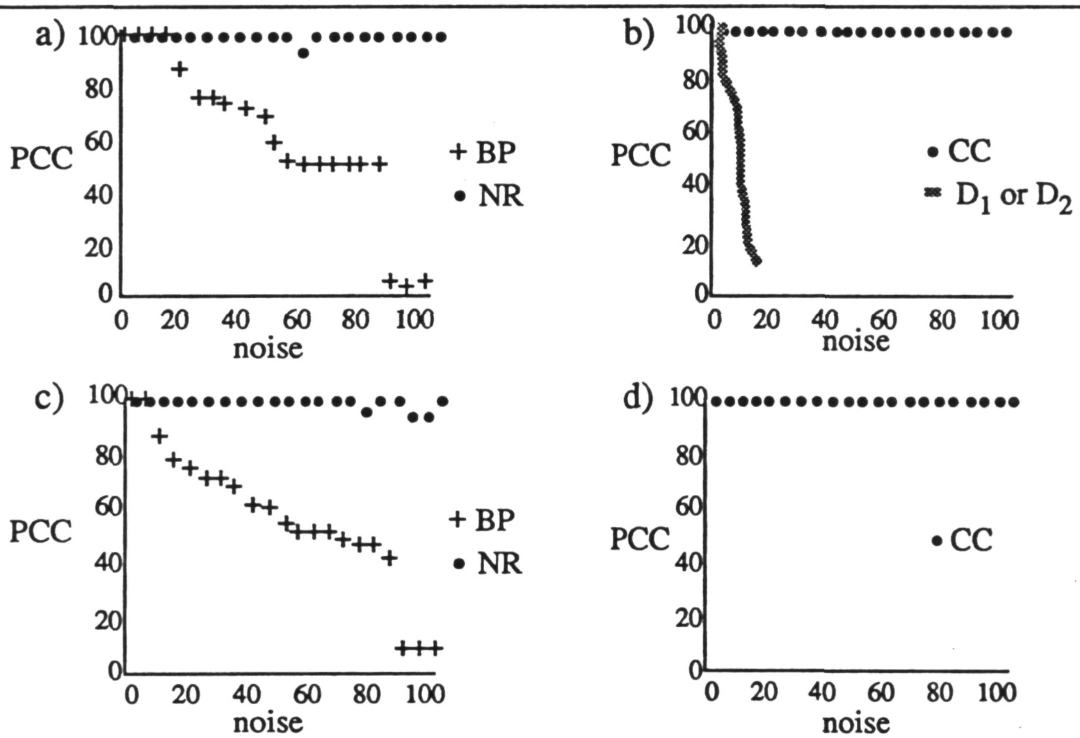


Figure 4. Probability of correct classification (PCC) as a function of additive noise percentage for back-propagation (BP), the neuron ring (NR), cross correlation (CC), and the non-fuzzy ring structure using non-fuzzy rules D_1 or D_2. Non-fuzzy performance is approximate. BP trained on spatial data ; spatio-temporal patterns produced by averaging Fourier data records. Top and bottom graphs pertain to the frigate FF 1041A and to the *Elizabeth*, respectively. Each trace is based on 5000 simulated ship encounters or trials.

the reverse situation for noise ramping down from 75% to 25% with increasing frequency. Thus, the effective correct classification percentage is 99.99% for these tests.

The format of the convexity test was simply to mix the signals then excite the network with the mixture. Work is still in progress to detect if the mix percentage

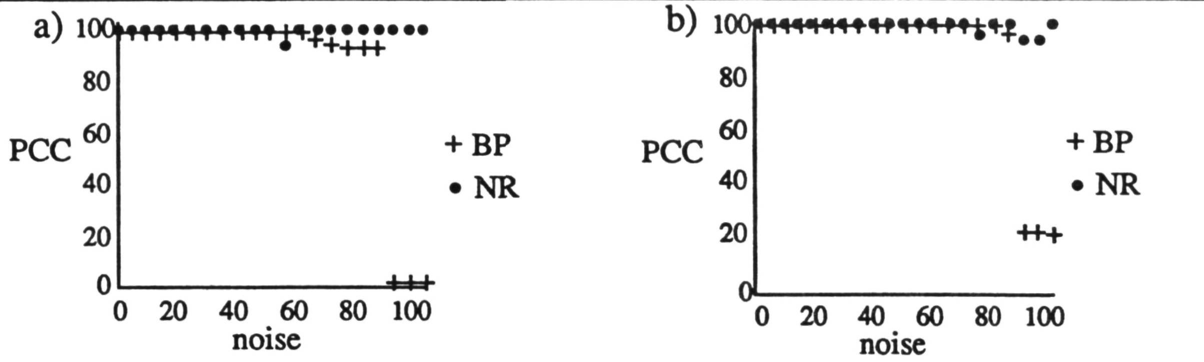


Figure 5. Performance of backpropagation and the fuzzy ring in uniform additive zero mean percentage noise for the frigate and boat 2. Training is with spatio-temporal patterns.

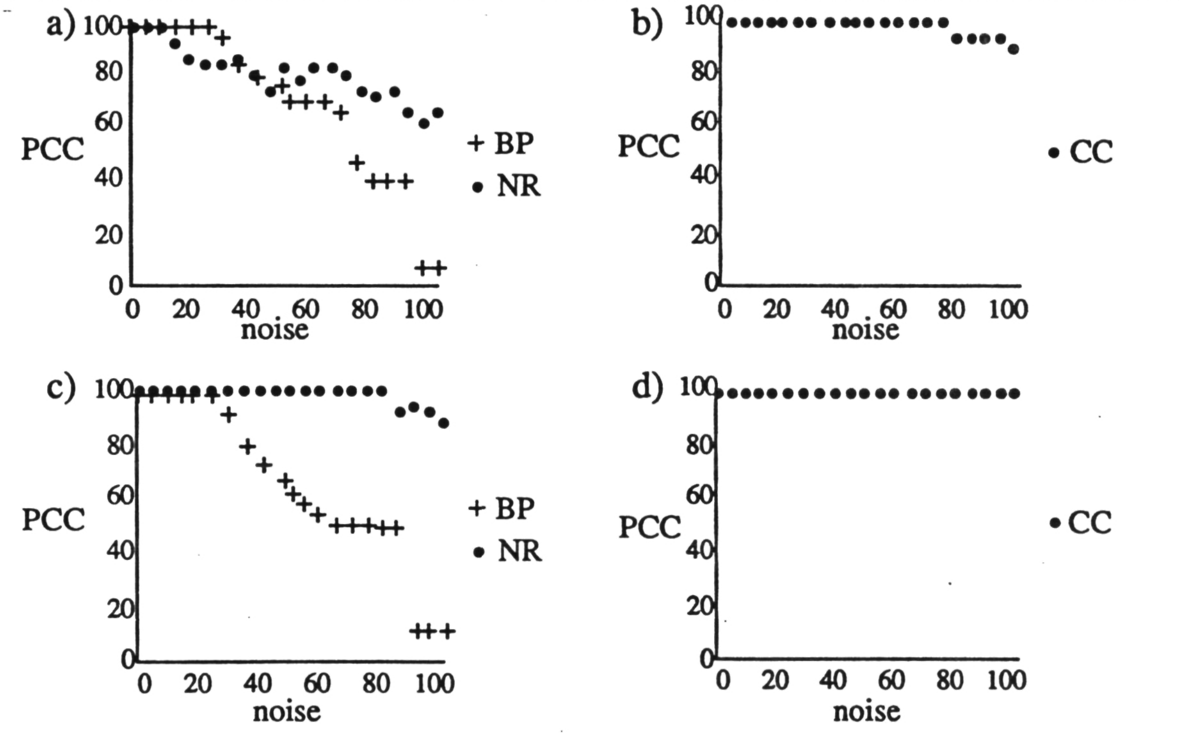


Figure 6. PCC as a function of additive noise. Signals shifted in time by several minutes. Top diagrams are for the frigate FF1041A and the bottom diagrams are for boat 2. BP training is spatial only.

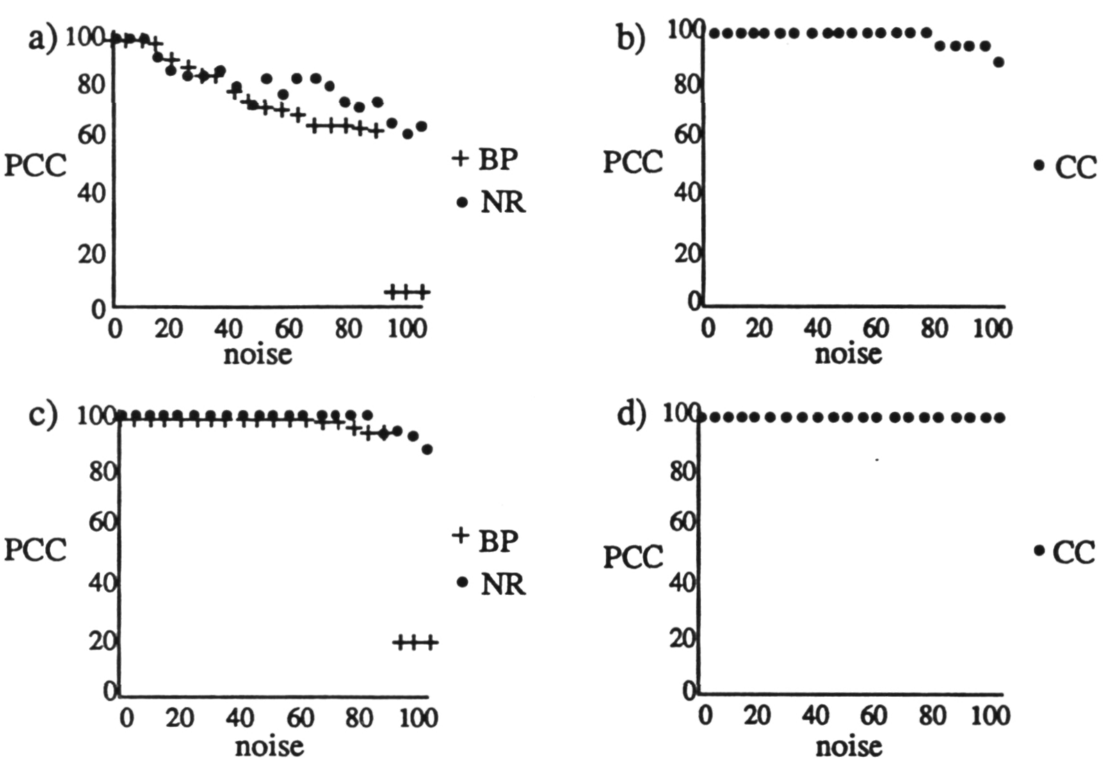


Figure 7

Figure 7. Probability of correct classification as a function of uniform additive noise. Time shift ~ 2 minutes. a) BP and NR for boat 2; b) cross correlation for boat 2. c) BP and NR for the frigate; and d) cross correlation for the frigate. Training is spatio-temporal.

propagates to the decision metrics discussed earlier.

SUMMARY

This study implies that the analysis of the historical activation record or activation field is more effective than using point estimates, at least for simple structures such as the neuron ring. We simulated over a quarter of a million ship encounters in the study; the graphs indicate typical performance. The breadth of the noise and signal characteristics lend credence to the thesis of this paper; namely, that excitation induces a possibility distribution on the neuron's activation. The total support for the ring's hypothesis is the sum of each neuron's possibility measure. This support is a better indicator of pattern class than those used before this study.

REFERENCES

1. Bellman, R., Kalaba, R., & Zadeh, L., Abstraction and pattern classification, Journal of Mathematical Analysis and Applications, **13**, 1-7, 1986.

2. Bendat, J.S. and A.G. Piersol. *Random data: analysis and measurement procedures*, second edition, John Wiley & Sons, New York, 1986.

3. Frank, M.J. On the simultaneous associativity of F(x,y) and x+y - F(x,y). Aequat. Math., **19**, 194-226, 1979

4. Goles, E. Lyapunov functions associated to automata networks. *in* Automata Networks in Computer Science, Theory and Applications, Princeton University Press, F.F. Soulie, Ed., 1987. pp 58-81.

5. Grossberg, S., A prediction theory for some nonlinear functional-differential equations: I, learning of lists, Journal of Mathematical Analysis & Application, **21**, 643-694, 1968.

6. Hecht-Nielsen, R., Nearest matched filter classification of spatiotemporal patterns, Applied Optics, **26**, 1892-1899, 1987.

7. Keller, J.M. and D.J. Hunt. Incorporating fuzzy membership functions into the perceptron algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence, **PAMI-7**, no.6, 693-699, 1979.

8. Kosko, B., Counting with Fuzzy Sets, IEEE Transactions on Pattern Analysis and Machine Intelligence, **PAMI-8**, no. 4, 556-557, 1986.

9. Parker, D., Learning logic, invention report, SBI-64, File 1, Office of technology licensing, Stanford University, 1982.

10. Rumelhart, D. E., McClelland, J. L., Eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vols 1 and 2, MIT Press, Cambridge, 1986.

11. Strat, T. Continuous Belief Functions, Proceedings, AAAI-84, 308 - 313, 1984.

12. Taber, W.R. An Imaging System for the Study of Fluorescent Protein Movement in the Microcirculation. Ph.D. Dissertation, Texas A&M University, 1984.

13. Werbos, P., Beyond regression: New tools for prediction and analysis in the behavioral sciences, Ph.D. Dissertation, Harvard University, 1974.

14. Yager, R.R. Toward a General Theory of Reasoning with Uncertainty. I: Nonspecificity and Fuzziness, International Journal of Intelligent Systems, **1**, no. 1, Spring, 1986.

15. Zadeh, L. A., Fuzzy Sets, Information and Control, **8** 338-353, 1965.

16. Zadeh, L. A., Test-score semantics for natural languages and meaning-representation via PRUF, Technical note 247, SRI International, 1981.

17. Zadeh, L. A., Making Computers Think Like People, IEEE Spectrum, Aug. 1984.

Fuzzy Logic and Neural Networks

by

W. R. Taber

General Dynamics
Electronics Division

San Diego, California

WHAT CAN FUZZY LOGIC

DO

FOR NEURAL NETWORKS ?

This paper describes the results of a study to find minimal neural structures that are able to recognize ships from underwater recordings.

We propose some architectural change to the neuron.

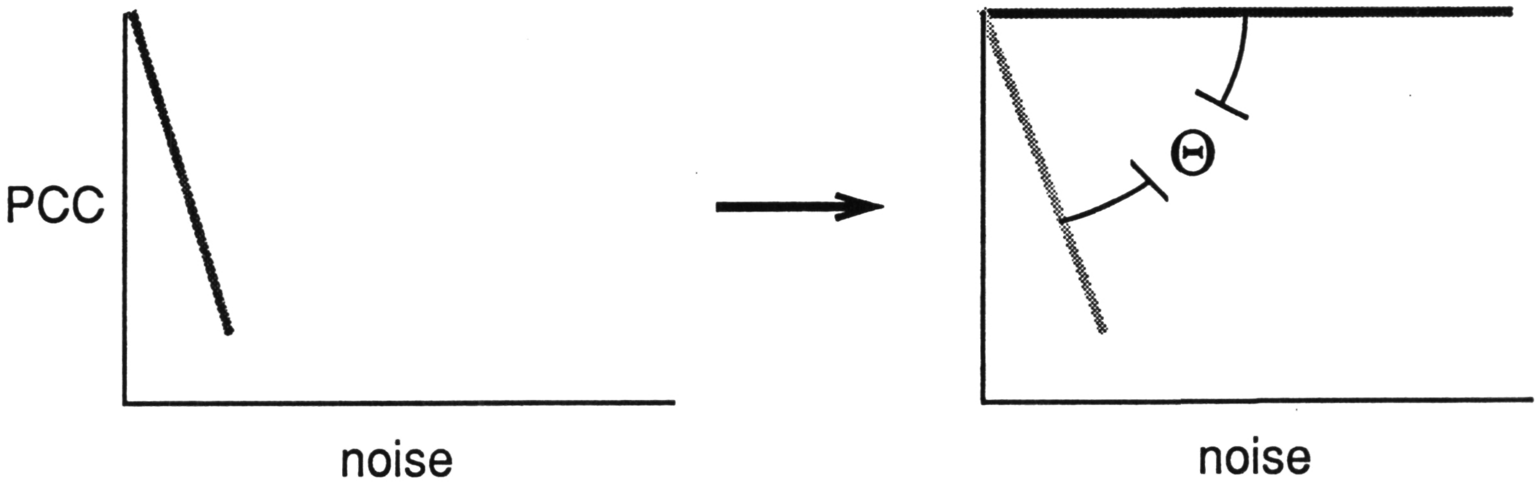
Experiments were designed to test the ability of neural networks to correctly classify ships. During the tests, it became clear that the avalanche worked as an extremely narrow band matched filter.

We went back to the basics and asked:

"What is the neuron doing?"

"Can we boost performance?"

$\uparrow$ Performance = $\downarrow \Theta$



AT THE HEART OF MOST NEURAL NETWORKS

DOT PRODUCT NEURON
PERFORMS

$$u \bullet v = x$$

u = prestored reference vector

v = unknown or test vector

x = nascent excitation scalar

$$F(x) \rightarrow a \in (0,1)$$

GENERAL DYNAMICS
ELECTRONICS DIVISION

WHY THIS WORKS

Suppose u, v are positive unit vectors on Hypersphere in R^n

$$\frac{u \cdot v}{||u|| \cdot ||v||} = \cos \phi$$

But $||u|| \cdot ||v|| = 1$ (unit vectors)

So $\cos \phi = u \cdot v \, \varepsilon \, (0,1)$

CORRELATION

$$R_{uv}(r\,\Delta\,t) = \frac{1}{N - r} \sum_{i=1}^{N-r} U_i \bullet V_{i+r}$$

for time delay = 0:

$$R_{uv} = \frac{1}{N} \sum_{i=1}^{N} U_i \bullet V_i$$

if N is constant over all patterns.

$$\therefore U \bullet V = R_{uv}$$

Activation by correlation

BASIC OPERATION

$$U \bullet V = x$$

$$\text{Activation} = F(x)$$

$$\text{Example: } F = \frac{1}{1 + e^{-x}}$$

What happens to information if $F(x) \neq x$?

a. We make-up spurious information

or

b. We discount good information

a. If we add spurious information (spurium?)

are we degrading performance ?

b. If we discount good information,

are we degrading performance ?

The problem domain is underwater acoustics

We would like a system to tell us what ship or kind of ship is in the water

We designed a series of experiments to find a system model

DATA CAPTURE

- Record ships with a hydrophone
- Filter the analog waveform at 8 KHz
- Digitize .6 seconds at 20 KHz
- 1024 pt Fourier transform on 20 time slots per ship
- Compute power spectra
- Train neuron rings for the pattern set

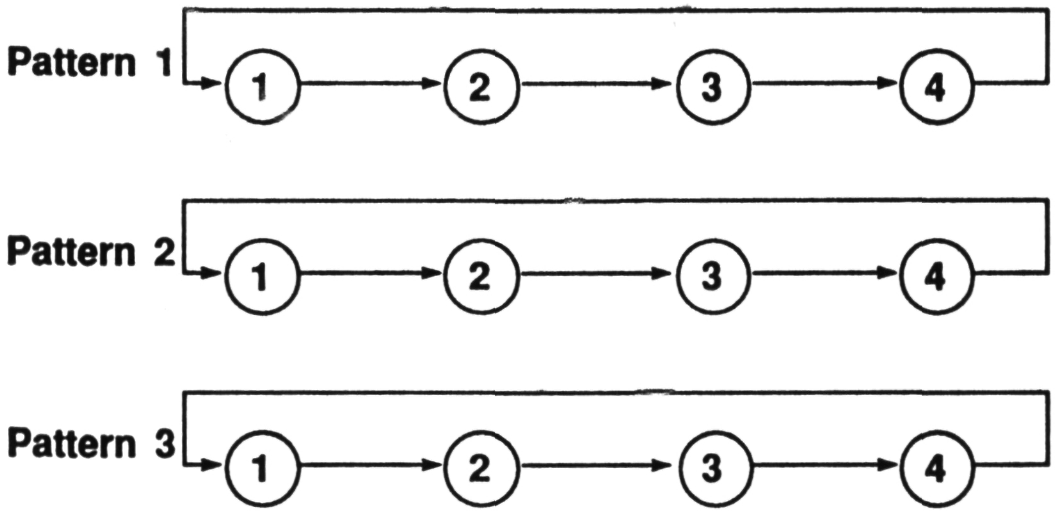
TEST

- Pick test signal
- Add noise to test
- Classify

NOISE TYPES

- Pseudo-white
- Ramp up with frequency
- Ramp down with frequency
- Time shift
- Convex combinations of existing signals

NEURON RING



GENERAL DYNAMICS
ELECTRONICS DIVISION

$$X_i^{new} = X_i^{old} + A[-a\, X_i^{old} + b\, I_1 + c\, I_2]^+$$

X_i^{new} = Activation for neuron i

X_i^{old} = Old activation

A = Attack factor

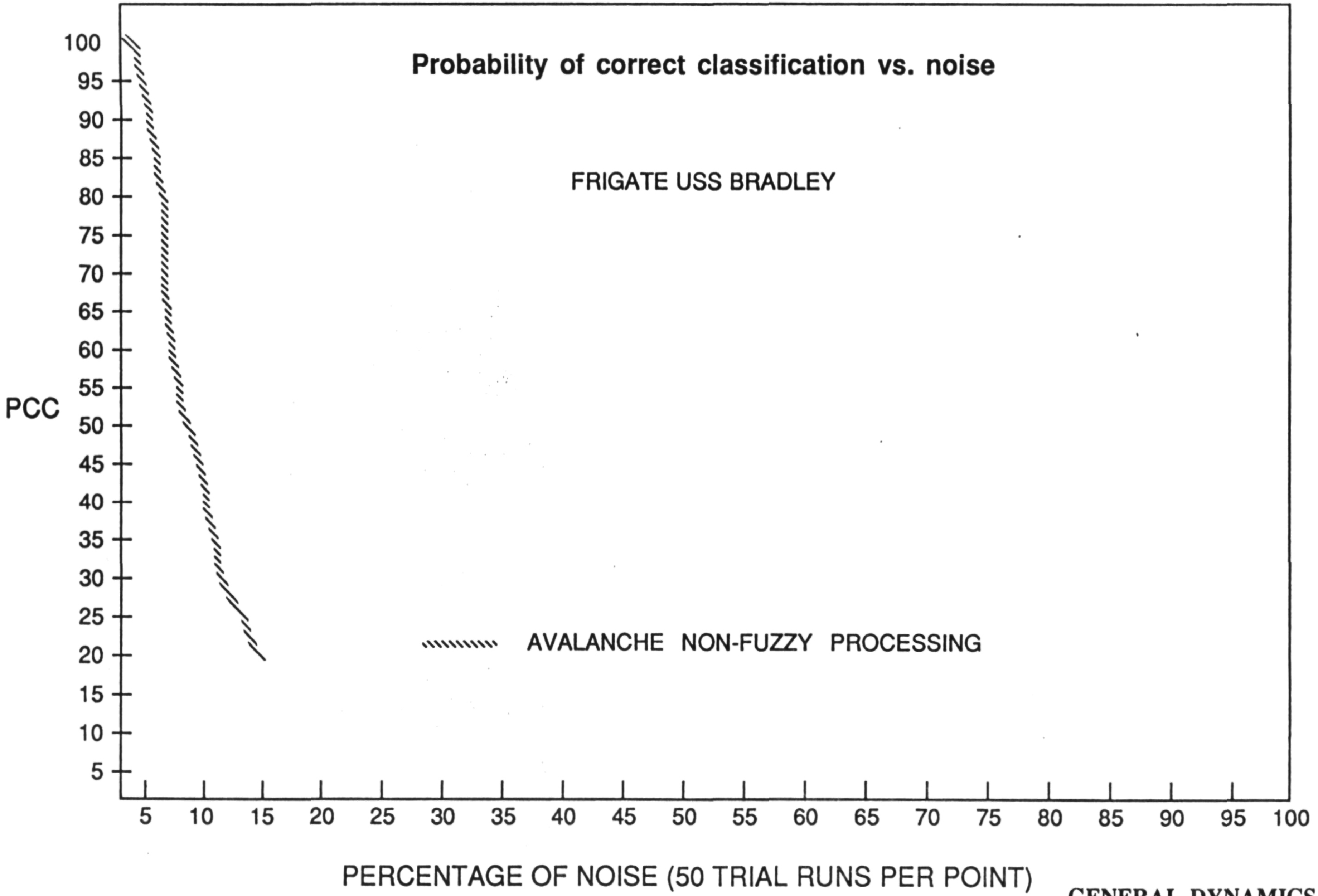
a = Decay constant for old activation

b = Gain for activation from previous neuron (decouple)

I_1 = Activation from previous neuron

c = Gain for dot product

I_2 = Dot product of pattern $\vec{Z}_i$ with unknown $\vec{U}_i$



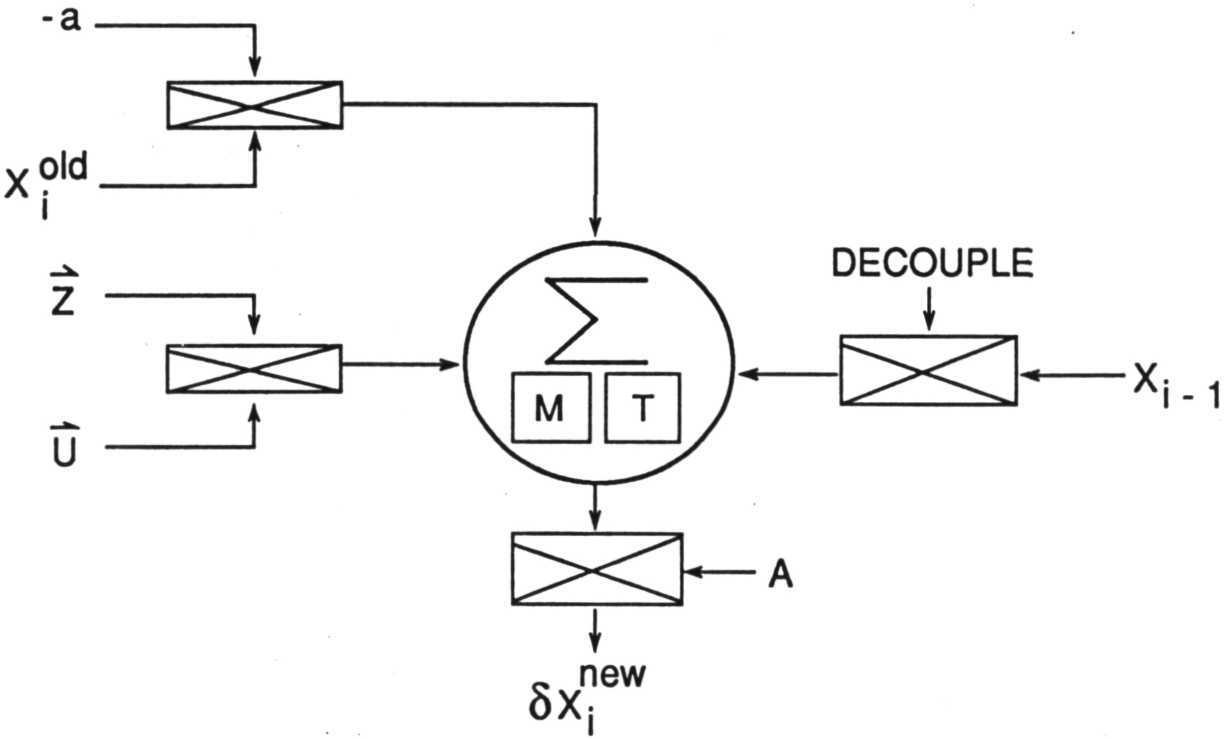
GENERAL DYNAMICS
ELECTRONICS DIVISION

<u>Consider</u> fuzzy sets and multi-valued logic.

- Distribution free
- Graded membership is an attractive idea

<u>Probability</u>	<u>Possibility</u>
P (e or ~e) = 1	Po (e or ~e) = max (e or ~e)
P (.8 or .2) = 1	Po (.8 or .2) = .8

This enables fuzzy processing



The Dot Product Neuron with Latches

LET'S ALLOW THE FOLLOWING F:

$$F(x) = \begin{cases} x & \text{if } 0 \le x \le 1 \\ 1 & \text{if } x > 1 \end{cases}$$

THEN WE MINIMIZE THE MODIFICATION OF INFORMATION

PROCEDURE

- Gate each pattern to every neuron
- Present all 20 vectors
- Decide which pattern it is $D_1 \ldots D_4$

D_1: The ring with the highest acceptable activation in its last neuron wins.

D_2: The ring with the first activation of 1 wins the competition.

$$D_3 = \mathrm{arc}\left(\bigcup_i \Sigma C_i\right)$$

$$D_4 = \mathrm{arc}\left(\bigcup_i FM_i\right)$$

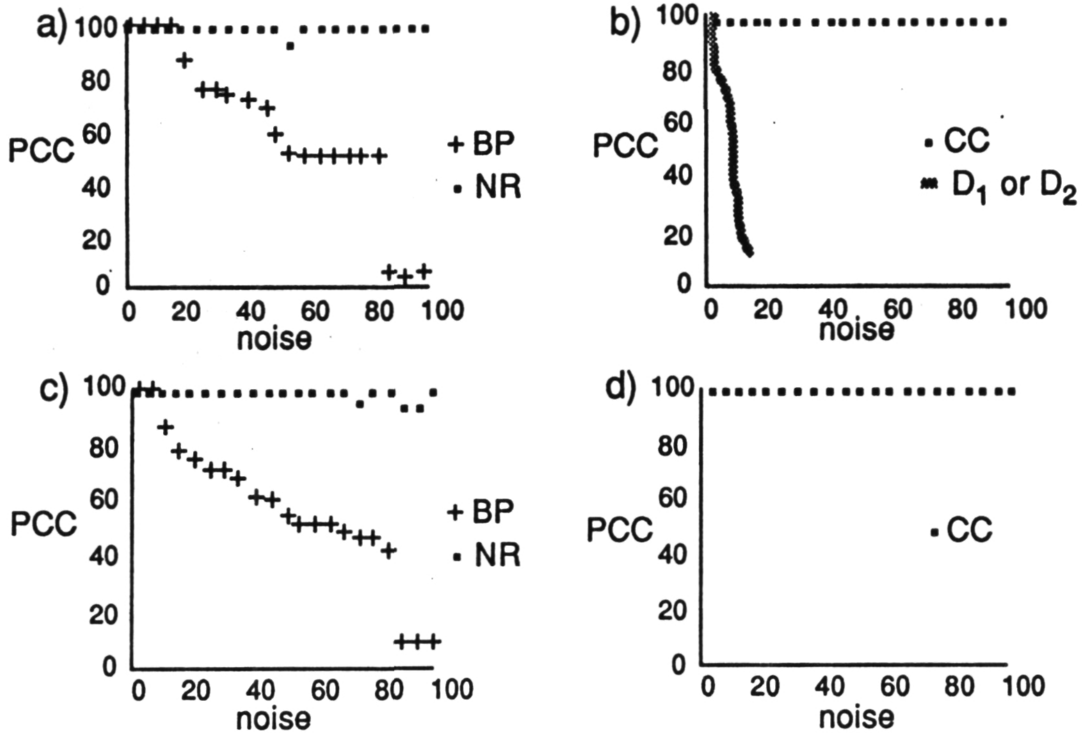


Figure 4. Probability of correct classification (PCC) as a function of additive noise percentage for back-propagation (BP), the neuron ring (NR), cross correlation (CC), and the non-fuzzy ring structure using non-fuzzy rules D_1 or D_2. Non-fuzzy performance is approximate. BP trained on spatial data only; spatio-temporal patterns reduced by averaging Fourier data records.
Top frigate; bottom boat 2.

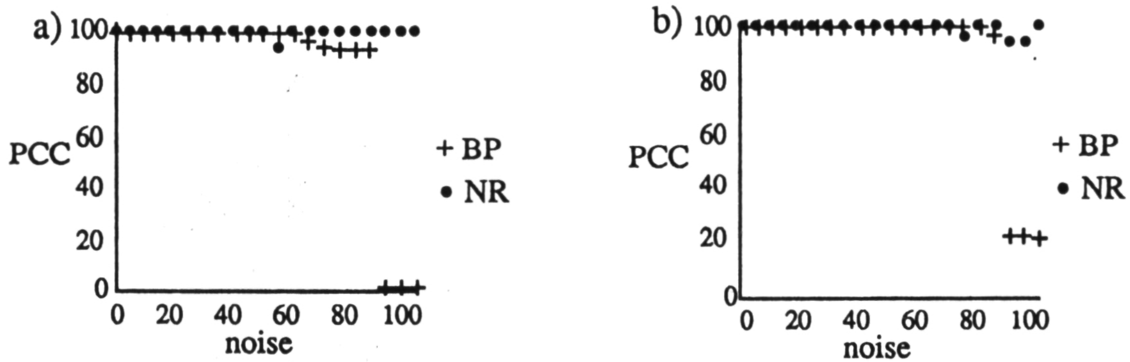


Figure 5. Performance of backpropagation and the fuzzy ring in uniform additive zero mean noise

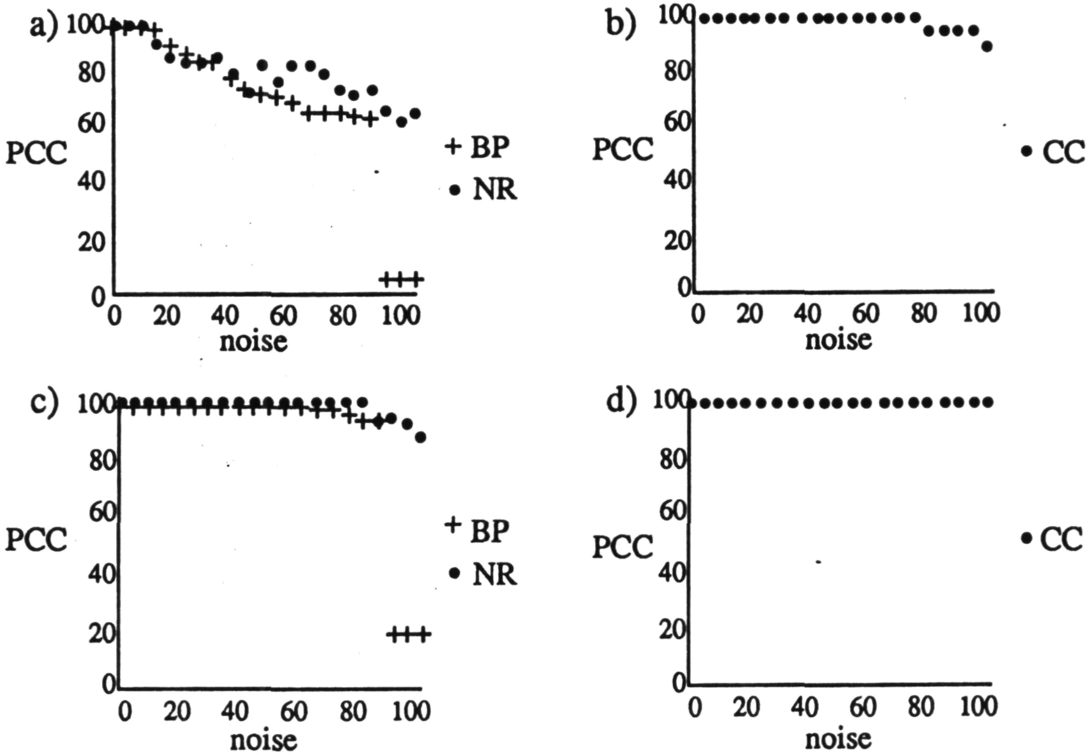


Figure 7

Figure 7. Probability of correct classification as a function of uniform additive noise. Time shift ~ 2 minutes. a) BP and NR for boat 2; b) cross correlation for boat 2. c) BP and NR for the frigate; and d) cross correlation for the frigate. Training is spatio-temporal.

FUZZY PROCESSING

1. Patterns induce a Possibility distribution on the neuron
2. Height of the distribution equates to possibility measure
3. ΣC of the heights $\longrightarrow$ classification

DECISION RULES

NON-FUZZY

D_1: The ring with the highest acceptable activation in its last neuron wins.

D_2: The ring with the first activation of 1 wins the competition.

FUZZY

$$D_3 = \mathrm{arc}\left(\bigcup_i \Sigma C_i\right)$$

$$D_4 = \mathrm{arc}\left(\bigcup_i FM_i\right)$$

ΣC = Sigma-count
arc = pre-image of the argument

STATISTICS FOR TEST FILE: SHIPS INPUT SIGNAL: Boat 2

WINDOW =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	ΣC
Boat 2	0	0	2	4	6	5	5	5	4	6	6	7	7	7	6	8	7	9	10	10	114
Boat 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Elizabeth	0	0	2	4	3	3	3	3	3	2	2	2	2	2	2	2	2	4	5	5	51
SEINER	0	0	2	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	14
FF1041A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FF1041B	0	0	2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	12
FFG41B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FFG41C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DREDGE	0	0	0	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	1	20
ZODIAC	0	0	2	4	5	5	5	4	4	4	4	4	3	3	3	3	3	3	4	6	69

CERTAINTY MEASURES FOR INPUT SIGNAL: Boat 2

TRAINING SIGNAL	CERTAINTY RATIO	FUZZY MEMBERSHIP
Boat 2	0.407	100
Boat 3	0.000	0
Elizabeth	0.182	45
SEINER	0.050	12
FF1041A	0.000	0
FF1041B	0.043	11
FFG41B	0.000	0
FFG41C	0.000	0
DREDGE	0.071	18
ZODIAC	0.246	61

CLOSEST MATCH FOR INPUT SIGNAL: Boat 2
CURRENT PERTURBATION PERCENT = 40
TEST NUMBER = 25

GENERAL DYNAMICS
ELECTRONICS DIVISION

SUMMARY

We fuzzified the neuron

$F(x) = x$ if $0 \le x \le 1$; 1 if $x > 1$; 0 if $x < 0$	graded membership
M latches maximum F(x); T latches time	possibility
ΣC of M yields hypothesis support	graded membership
Θ approaches 0	near optimal performance

GENERAL DYNAMICS
ELECTRONICS DIVISION

NOTES

59-63
ABS. ONLY

N91-71369

Douglas L. Reilly, Ph.D.

Nestor, Inc.
Providence, Rhode Island

Dr. Reilly is a summa cum laude graduate of Georgetown University, with a B.S. in physics. He received his M.S. and Ph.D. degrees in physics at Brown University, doing research in the field of neural networks. His thesis was entitled "A Neural Model for Category Learning." From 1980 to 1983, Dr. Reilly was a postdoctoral research fellow and assistant professor at the Center for Neural Science at Brown. In 1983, he joined Nestor, Inc., as its first employee and vice president of research and development. Dr. Reilly was instrumental in establishing Nestor's research and development office, and under his direction, the company has developed an adaptive pattern recognition technology based upon neural network principles - applying that system to problems in character recognition, speech recognition, object recognition, and risk assessment in financial services.

ADAPTIVE PATTERN RECOGNITION USING A MULTINEURAL NETWORK LEARNING SYSTEM

Abstract

A learning system composed of multiple neural networks and present examples of its application to problems in adaptive pattern recognition is discussed. The system makes use of multiple restricted Coulomb energy (RCE) networks that are powerful pattern classification subsystems, able to dynamically learn to separate nonlinearly-separable pattern classes in feature space, as well as to estimate class probabilities in nonseparable portions of the feature space. A controller integrates the responses of these various multiple neural networks to produce an overall system response. Additionally, the controller determines the training signals directed to the various component networks of the system to ensure that networks train to make the decisions for which they are best suited. Results of applying the system to problems in character recognition, industrial parts inspection, and decision support for risk analysis will be reviewed.

NOTES

530-63
ABS. ONLY

N91-71370

James C. Bezdek, Ph.D.

Boeing Electronics
Company
Seattle, Washington

Dr. Bezdek received his B.S. in civil engineering from the University of Nevada (Reno) in 1969, and his Ph.D. in applied mathematics from Cornell University in 1973. Currently, he is the director of the Information Processing Lab at the Boeing Electronics High Tech Center. Dr. Bezdek is the past president of NAFIPS, the current president of IFSA, and the editor of the International Journal of Approximate Reasoning. His research interests include pattern recognition, expert systems, information retrieval, and optimization.

KNOWLEDGE REPRESENTATION BY LINGUISTIC TRANSITIVE CLOSURES OF TRAPEZOIDAL FUZZY NUMBERS

Abstract

We present a theory for the representation and manipulation of uncertainties that might be supplied by an expert (or team thereof) about object-pair relationships in some knowledge domain. We propose a theory based on the representation of relational knowledge by semantic term sets and trapezoidal fuzzy numbers. The extended max - (*) linguistic transitive closure (LTC) is offered as a means for consistency enforcement and completion of partial knowledge in the relational network. Theorems are given that provide conditions for the existence and uniqueness of the LTC under three (extended) T-norms. We present an algorithm for computing each LTC and exhibit a number of features of this method through numerical examples.

NOTES

71

S21-63
ABS ONLY

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

N91-71371

Bill P. Buckles, Ph.D.

Tulane University
New Orleans, Louisiana

Dr. Buckles is an associate professor of computer science at Tulane University. He received his B.S. in mathematics, M.S. degrees in computer science and operations research, and Ph.D. in operations research from the University of Alabama in Huntsville, 1981.

RELATION BETWEEN UNCERTAINTY REPRESENTATION IN DATA BASES AND RULE-BASED SYSTEMS

Abstract

Uncertainty in a rule (if A, then B) arises from its deductive validity, the preciseness of the antecedent A, and the proximity of A to the data to which it is matched. The latter two causes of uncertainty are both related to the data and its representation. Uncertainty in data represented in data bases takes the form of null values, range values, nonatomic values (e.g., embedded relations), and various representations based on fuzzy set theory. There are similarities between the instantiation of the terms in a query and the action of matching rule antecedents. The latter is further complicated by the unification process, which, in some ways, resembles evaluation of transitive queries. These and other correspondences (and differences) are examined.

NOTES

S52-61

15322

N91-71372

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

James J. Buckley

University of Alabama
Birmingham, Alabama

Dr. Buckley received his Ph.D. in applied mathematics from Georgia Tech University in 1970. From 1970 to 1976, he was an assistant professor at the University of South Carolina, and since 1976, he has been an associate professor of mathematics at the University of Alabama at Birmingham. Dr. Buckley's research interests are in fuzzy sets, mathematical programming, control, decision theory, economics, game theory, and artificial intelligence. He is also an associate editor of the ORSA Journal on Computing.

LINEAR FUZZY CONTROLLER

Abstract

We consider a process controlled by a controller described by an n-th order linear ordinary differential equation toward its target output. As a special case, the controller is a proportional-integral-derivative (PID) controller. We show how to construct a linear fuzzy controller that gives precisely the same control as the PID controller. It is speculated that nonfuzzy controllers and fuzzy controllers may coincide on an unsuspectingly large class of control problems.

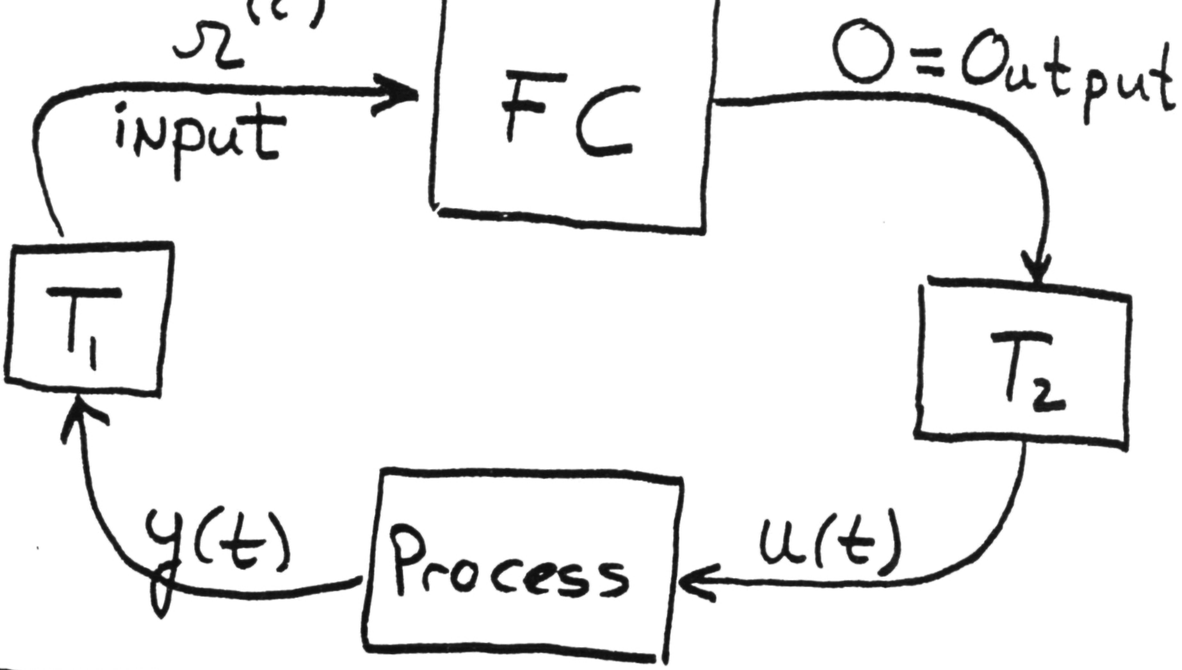
NASA Conference
MAY '88

Fuzzy Controller Theory

1. Linear FC.
2. Linear Control Rules.

J. J. Buckley
UAB

and H. Ying
Carraway



$\boxed{T_1}$ s = set point.

$y^{(i)}(t) = i^{th}$ derivative Error

$= y(t) - s$, $0 \le i \le n$.

$y^{(0)}(t)$ = Error.

<u>Scale</u>

$r^{(i)}(t) = c_i \, y^{(i)}(t) \in [-1, 1]$,

$0 \le i \le n$, $t \ge 0$.

input to FC.

$\boxed{T_2}$

$$\bigcirc \xrightarrow{\text{defuzzify}} \delta(t) \in [-1, 1]$$

Output

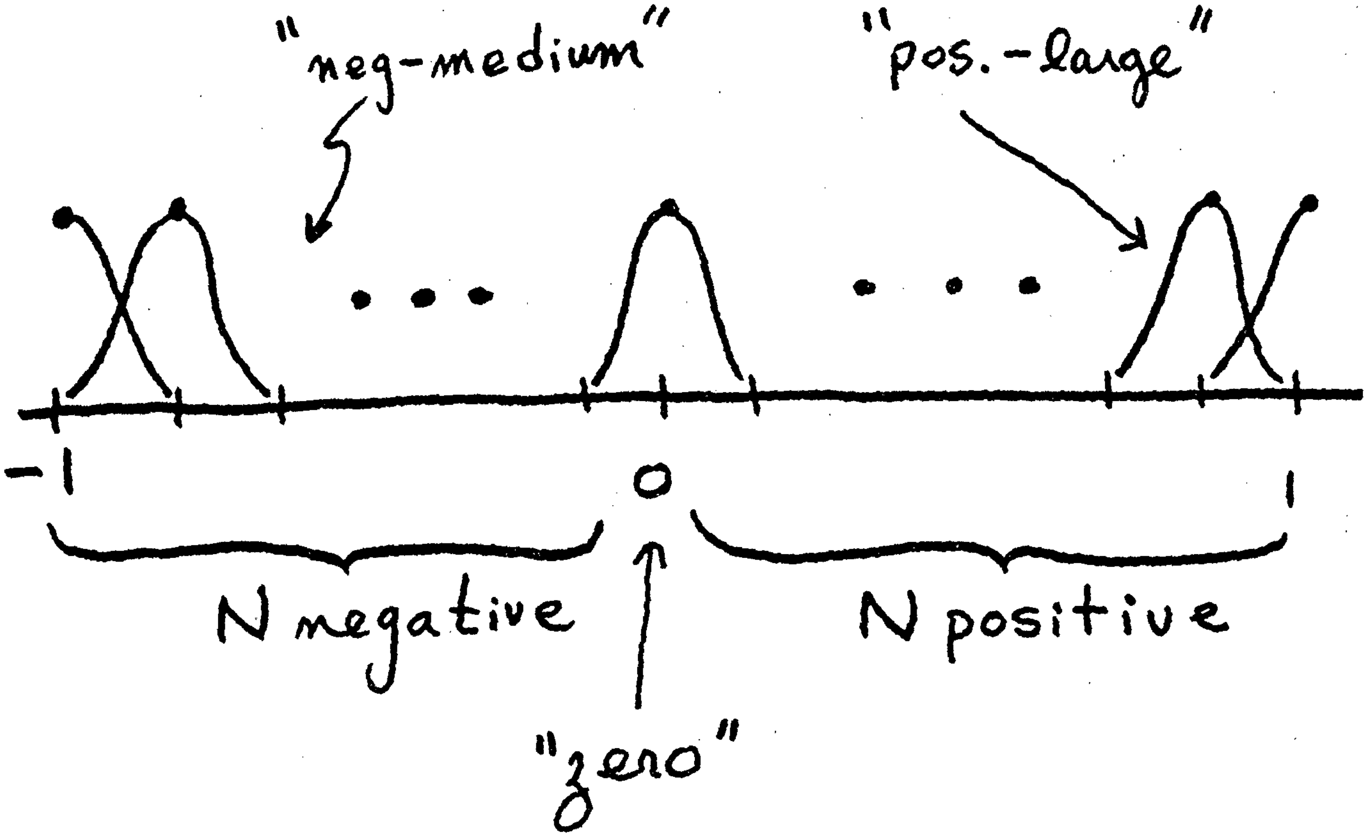
$$u(t) = u(t-\Delta) + \delta(t)\Delta, \quad t = \Delta, 2\Delta, \cdots$$

$\boxed{\text{FC}}$

1. Fuzzify — fuzzy numbers for Linguistic variables.
2. Rules, and their evaluation.
3. Defuzzify.

1. Fuzzy Numbers:

For each input $r^{(i)}$



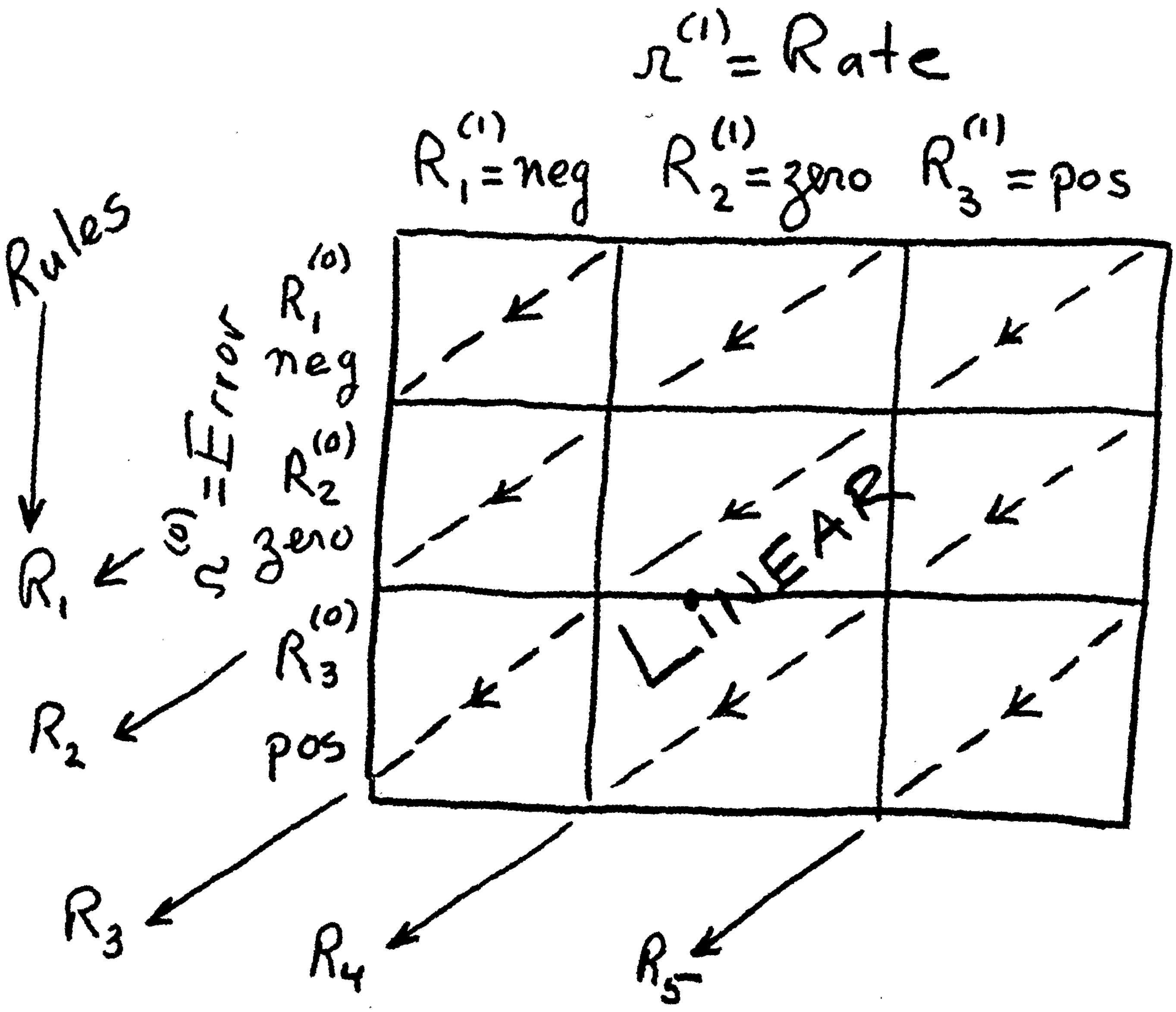
$2N+1$, $N \geq 1$, fuzzy numbers.

Equally spaced. Can be triangles, trapezoids, "normal", ⋯.

<u>Named</u> $R_j^{(i)} \leftarrow$ input, $1 \leq j \leq 2N+1$.

2. Rules:

$n = N = 1$. Then generalize!



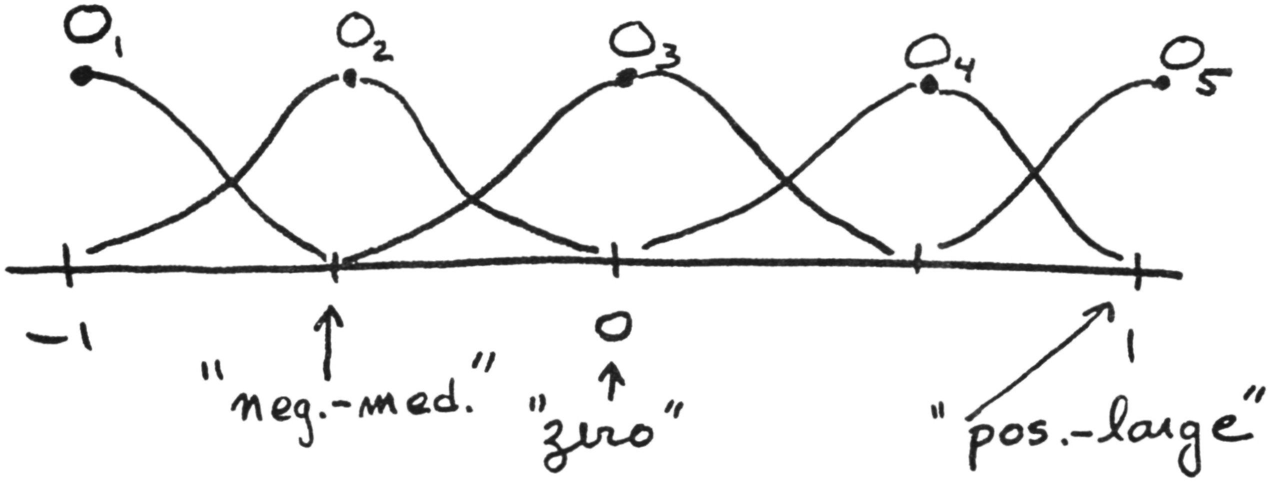
Rule $R_i \rightarrow [O = O_{6-i}]$.

$1 \le i \le 5$

Output

fuzzy #

$$O = \text{Output} = \left\{ \frac{\Delta_1}{O_1}, \ldots, \frac{\Delta_5}{O_5} \right\}$$



R_2: If [(Error = Neg) AND (Rate = zero)]

OR [(Error = zero) AND (Rate = neg)],

then $O = O_4$.

Evaluate all rules:

value LHS $R_i = \Delta_i =$

membership value of O_{6-i}.

$$\Delta_1 = T\left(\mu(r^{(0)} \mid \mathrm{Neg}), \mu(r^{(1)} \mid \mathrm{Neg})\right)$$

$$\Delta_2 = C\Big(T\left(\mu(r^{(0)} \mid \mathrm{neg}), \mu(r^{(1)} \mid \mathrm{zero})\right),$$
$$T\left(\mu(r^{(0)} \mid \mathrm{zero}), \mu(r^{(1)} \mid \mathrm{Neg})\right) \Big)$$

$\vdots$

T = any t-norm.
C = any co-t-norm.

Need not be the same from rule to rule.

---

for Δ_1, T can be MIN,
for Δ_2, T can be product,

$\vdots$

3. Defuzzify:

Defuzzified $O = \delta \in [-1, 1]$

(a) cv_i = central value of O_i

$$\delta_1 = \frac{\sum_{i=1}^{K} \Delta_i \, cv_{K-i+1}}{\sum_{i=1}^{K} \Delta_i}$$

K = # of rules.

In general, $K = (n+1)(2N) + 1$.

(b) All "reasonable" defuzzifiers.

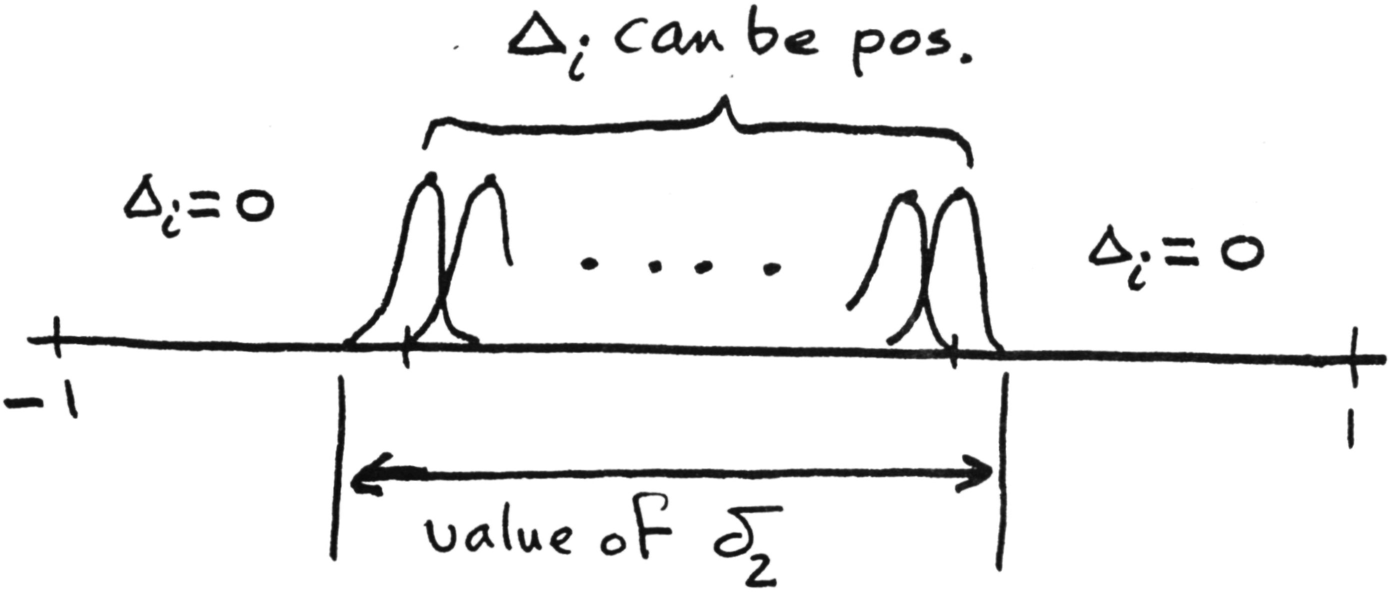
Contains: (i) δ_1,

(ii) center of gravity,

(iii) max membership,

⋮

Output = 0



No specific value of δ_2 need be given for main results.

$\delta_2 \in \bigcup \{$ supports of O_i whose Δ_i can be pos. $\}$.

Results:

let $\mathcal{L} = -\sum_{i=0}^{n} \Omega^{(i)} \big/ (n+1)$.

1. Linear FC

Δ fuzzy numbers, T = prob. AND, C = Lukasiewicz OR

$\Rightarrow \delta_1 = \mathcal{L}$ all n, N.

Always linear!

PI, PID for $n = 1, 2$.

Refs:

1. Siler and Ying "Fuzzy Control Theory: Linear Case" FSS. Submitted.

Ideas for Linear FC, using different fuzzy logic to evaluate rules, etc. Above result ($\delta_1 = \mathcal{L}$) generalizes one of their results.

2. J.J. Buckley and H. Ying "Linear FC, it is a Linear Non-fuzzy Controller", <u>IJMMS</u>. Submitted. (Proof $\delta_1 = \mathcal{L}$)

Note

$$\delta_1 \overset{\text{linear}}{\equiv} \sum \Delta_i \, cv_{K-i+1}$$

bec. here

$$\sum \Delta_i = 1.$$

2. Linear Control Rules.

$\delta_2 \longrightarrow \mathcal{L}$ as $N \longrightarrow +\infty$.

Any fuzzy numbers, any T and C, any "reasonable" defuzzifier, all n.

Ref:

1. Buckley and Ying "

Automatica. Submitted.

ⓐ So:

$$\delta_2 \approx PI, PID, \cdots (N \text{ large}).$$

ⓑ <u>Rate of convergence</u>:

$$|\delta_2 - \mathcal{L}| \leq \frac{c}{N}, \; c = ?$$

Some results.

ⓒ "linear" rules sufficient but not necessary. N and S condition on rules so that $\delta_2 \to \mathcal{L}$ as $N \to +\infty$ is unknown!

ⓓ <u>Note</u> # Rules $\to +\infty$ as $N \to +\infty$.

(e) $\delta = F(\Omega^{(0)}, \ldots, \Omega^{(n)})$

find F for small N.

Some results! How nonlinear is it?

<u>See also</u>:

Buckley "Fuzzy vs Non-Fuzzy Controller", <u>FSS</u>. Submitted.

(f) At other extreme from $N \rightarrow +\infty$ is

2 fuzzy numbers, 3 rules

<u>See</u>: Ying, Siler and Buckley, "Fuzzy Control Theory: a Nonlinear Case", <u>NASA Conference</u>.

Also "Expert FC"

I : Theory → FSS. Submitted.
II : Output Strategies → FSS. Sub.
III : Combined Input and Output Strategies → in preparation.
IV : Overall Strategies. Next!

By J.J. Buckley and H. Ying

Fuzzy goals for rise-time, overshoot, ...

y_i = value of fuzzy goal =

H_i(scaling constants, δ, rules, fuzzy numbers, ...)

$i = 1, 2, \dots$

Objective

$$\max(y_1, y_2, \ldots)$$

Subject to : ____________

However H_i unknown!

Globally optimal FC.

Decision Theory approach.

Based on our Fuzzy Expert System FLOPS.

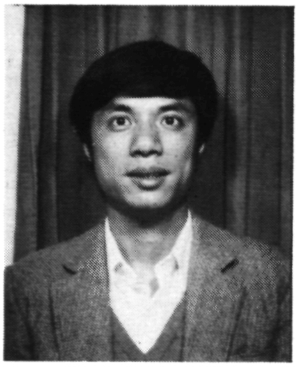
NOTES

73

5233-61

18363

N91-71373 p.1



Hao Ying, M.S.E.

Kemp-Carraway Heart Institute
Birmingham, Alabama

Mr. Ying received his masters degree in electrical engineering from the Department of Electrical Engineering and Computer Science, Textile University, Shanghai, China, in 1984. From 1984 to 1986, he was an instructor in the same department at the university. He is a Ph.D. student in the Department of Biomedical Engineering at the University of Alabama at Birmingham, and he is also a research fellow at Carraway Methodist Medical Center, Birmingham. Mr. Ying's research interests are in fuzzy sets, fuzzy control theory and its applications, expert systems, and artificial intelligence.

FUZZY CONTROL THEORY: A NONLINEAR CASE

Abstract

We prove theoretically that a nonlinear fuzzy controller is a nonfuzzy proportional-integral-derivative (PID) controller with proportional gain, integral constant, and derivative constant changing with error, rate change of error, and rate change of error rate about a setpoint of a process. The nonlinear fuzzy controller consists of the following parts:

1. The linear defuzzification algorithm

2. The linear fuzzy control rules

3. Zadeh's AND and OR fuzzy logics for evaluating the fuzzy control rules

4. The nonlinear defuzzification algorithm

The nonlinear fuzzy controller is a linear fuzzy controller which is precisely equivalent to a nonfuzzy PID controller if the linear defuzzification algorithm is used instead of the nonlinear one listed above.

The results of computer simulation reveal that the control performances of the nonlinear fuzzy controller and a nonfuzzy PID controller are almost the same if a linear process is controlled. However, the nonlinear fuzzy controller can control some nonlinear processes much better than a nonfuzzy PID controller does.